Mundo Spectrum

LOS MEJORES LISTADOS PARA TU MICRO

EDITA: 

·año·I·numero·I·

mundo precio 200 pesetas PVP IVA INC

LIST APPLE GLOTON

lunar car

TRUCOS

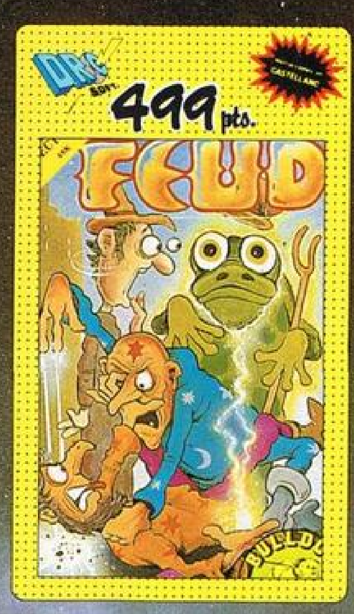
SUPER MUELLE

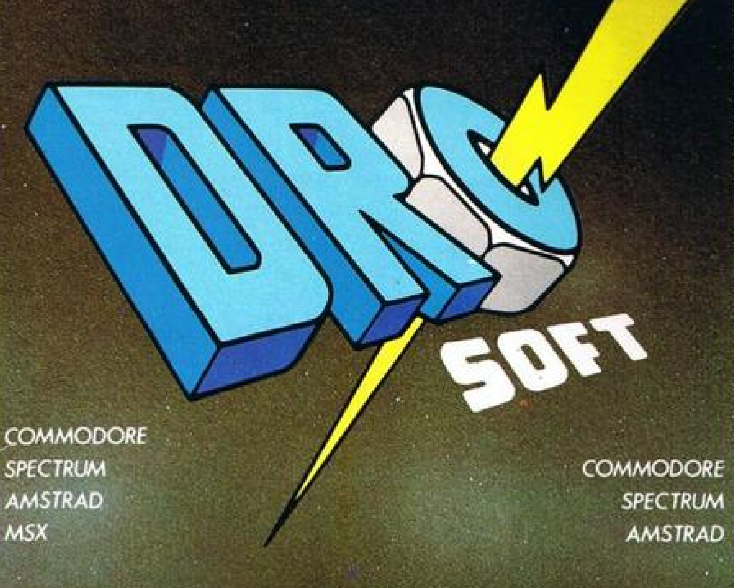
MISTER HUEVO

DISCIPLE

NOVEDAD

PVP CANARIAS 200 PTS







COMMODORE
SPECTRUM
AMSTRAD
MSX

FEUD
Con tu sabiduría puedes encontrar los ingredientes de las POCIONES MAGICAS, mientras recitas los conjuros de tu maligno enemigo LEANORIC.

COMMODORE
SPECTRUM
AMSTRAD

NINJA
Eres un NINJA, demuestra tu destreza luchando contra los mejores Budokas en un combate a MUERTE.



499 pts.

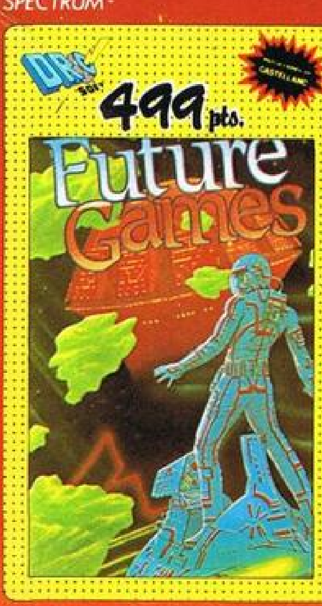
SERIE M.A.D. 699 pts.

COMMODORE
SPECTRUM
AMSTRAD
MSX

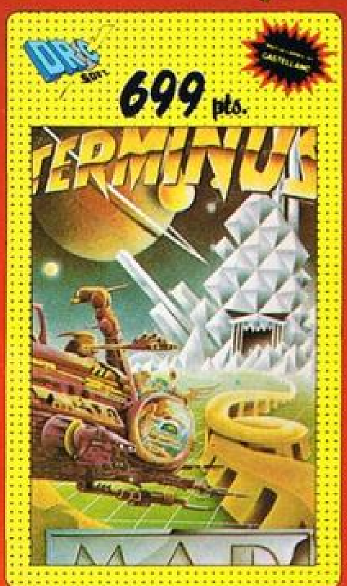


HYPERBOWL
En el año 3.600 el hockey sobre hielo ha alcanzado el nivel de máxima tecnología. Diez clases diferentes de naves, se enfrentan en un campo metálico...

SPECTRUM

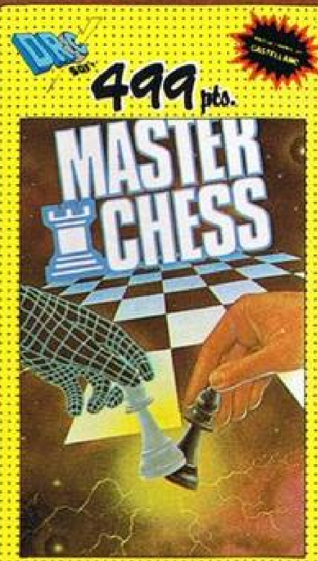


FUTURE GAMES
Juega tu libertad contra la muerte. Si ganas, la consigues, si pierdes, desde el lugar donde estés, ya no te importará.



TERMINUS
Un grupo de gamberros, que siembra el pánico por donde pasa, ha decidido rescatar a su líder del planeta prisión TERMINUS.

COMMODORE



MASTER CHESS
Tu ordenador es tu mejor contrincante. Piensa más rápido que tu, pero tu puedes jugar mejor y ganarle. ¡INTENTALO!

SPECTRUM
AMSTRAD
MSX



Y otros 120 juegos mas.

MASTERTRONIC

EDITORIAL

Bienvenidos a este primer número del Mundo Spectrum. Como podréis observar, el grueso de nuestra revista consta de programas listados, y esto es así porque pensamos que la mayoría de los usuarios del Spectrum debemos en buena parte nuestros conocimientos sobre informática al "muy noble" arte de teclear programas. De forma que nuestro empeño estará centrado en ofrecer un amplio repertorio de programas de calidad que nos permitan seguir aprendiendo de la manera más sensilla, útil y divertida posible.

En este número hemos seleccionado cuatro divertidos juegos así como una utilidad en Código Máquina y los hemos acompadado de unos amplios comentarios, que os permitirán saber línea a línea, qué es lo que estamos haciendo y porqué.

Además de un par de interesantes trucos, hemos incluido un artículo sobre la memoria del Spectrum, como elemento de repaso y estudio y otro sobre el novedoso Disciple, un producto que supone un notable avance en el tema de la ampliación de prestaciones del Spectrum.

No queremos terminar sin deciros que estamos abiertos a todo tipo de consultas y sugerencias que deseéis hacernos. Pretendemos además que esta relación sea fluida y constante. Ahora bien, no podemos comprometernos a responder por correo, por lo que en el próximo número abriremos una sección para responder o aclarar cualquier tema solicitado.

Por otro lado, también podréis contar con una sección de intercambio directo entre lectores, por lo que desde ya, esperamos vuestros anuncios.



RECOMENDACIONES PARA TECLEAR UN PROGRAMA

Para aquellos que no estéis muy acostumbrados a teclear programas, os daremos a continuación una serie de instrucciones o pasos a seguir para que os resulte más sencillo y las probabilidades de que se produzcan errores en la ejecución de los programas sean menores:

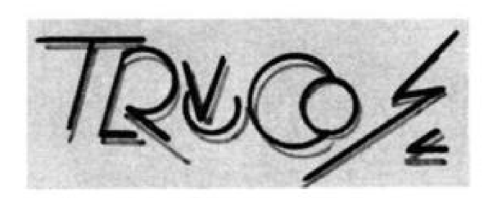
1. Como muchos ya sabréis, las líneas que después de su número correspondiente tienen una instrucción REM o un apóstrofe, no son consideradas por el programa sino que únicamente sirven como comentario aclaratorio para el programador y por lo tanto no es necesario que las copieis completas cuando aparezcan en nuestros listados, en cambio si es conveniente que pongáis el número de línea y la instrucción REM o ' (ejemplo 200 REM o 200') para evitar que puedan producirse errores al ejecutarse en el programa alguna instrucción de salto (GOSUB, GOTO, etc.)

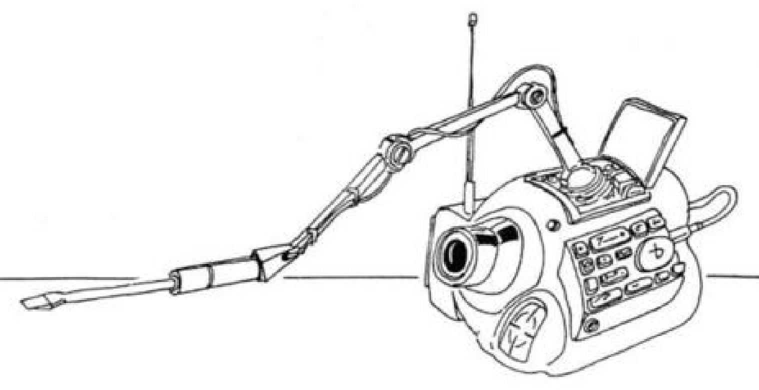
2. Es fundamental que los números de línea los copiéis exactamente como figuran en el programa original.

3. Es conveniente, sobre todo en programas largos, salvar periódicamente a cinta o disco la parte que llevemos tecleada del listado antes de probar su funcionamiento para evitar la pérdida completa del programa en caso de que exista un error en el código máquina o bien se produzca un corte accidental de corriente, etc.

Dichas grabaciones periódicas es aconsejable no efectuarlas sobre la copia anterior o con el nombre en caso de hacerlas en disco.

Mundo Spectrum es una publicación del Grupo Editorial SYGRAN S.A. Polig. Ind. Valdonaire. C/ Apolonia Hernández. HUMANES (Madrid). **Director:** Angel Herrero Fernández. **Director Adjunto:** Andrés Franco. **Director Técnico:** Luis Sanguino. **Coordinador Software:** Angel García. **Secretaría Redacción:** Mercedes Matons. **Colaboradores:** Antonio García, Angeles Guindal, Carmen Torres, J. Vicente Pons. **Diseño e Ilustraciones:** Javier Caballero. **Publicidad:** GENESIS. Tomás López, 3-6°. 28009 Madrid, Tel. 401 77 54 **Fotocomposición:** Fermar C/. Sílfide 10. **Imprime:** Gráficas Osiris S.A. Brañuelas, 29. Distribuye: G.M.E., plaza Castilla, 3. Madrid. Depósito Legal: M-1 31875 1987.





Os ofrecemos a continuación un par de trucos que os permitirán aprovechar de una forma más práctica, el teclado de vuestro Spectrum.

MOVIMIENTO CONTINUO

Para producir un movimiento continuo, vamos a utilizar la variable del sistema LAST K (Last Key), localizada en la posición de memoria 23560, y que almacena el valor de la tecla pulsada últimamente. De forma que este valor no cambie hasta que no se pulse otra tecla distinta.

El valor que contiene, es el código ASCII de la tecla pulsada. ¡¡¡Prueba el ejemplo y verás!!!

```
 10 REM ************************
 20 REM * MOVIMIENTO CONTINUO *
 30 REM ************************
 40 REM
 50 LET X=10
 60 LET Y=16
 70 LET T=PEEK 23560
 80 IF T=81 THEN LET X=X-1
 90 IF T=65 THEN LET X=X+1
100 IF T=79 THEN LET Y=Y-1
110 IF T=80 THEN LET Y=Y+1
120 IF X<0 THEN LET X=21
130 IF X>21 THEN LET X=0
140 IF Y<0 THEN LET Y=31
150 IF Y>31 THEN LET Y=0
160 PRINT AT X,Y;"*"
170 BEEP .05,10
180 PRINT AT X,Y;" "
190 GO TO 70
```

LECTURA DEL TECLADO CON IN.

Si necesitas leer la pulsación simultánea de varias teclas deberás emplear la función IN como sigue:

LET T = IN semifila: IF T > 191 THEN T = T - 64
IF T = 191 - 2 ntecla THEN...

donde:

Semifila dirección de la semifila que contiene la tecla a reconocer. Las tienes listadas en la tabla que se acompaña.

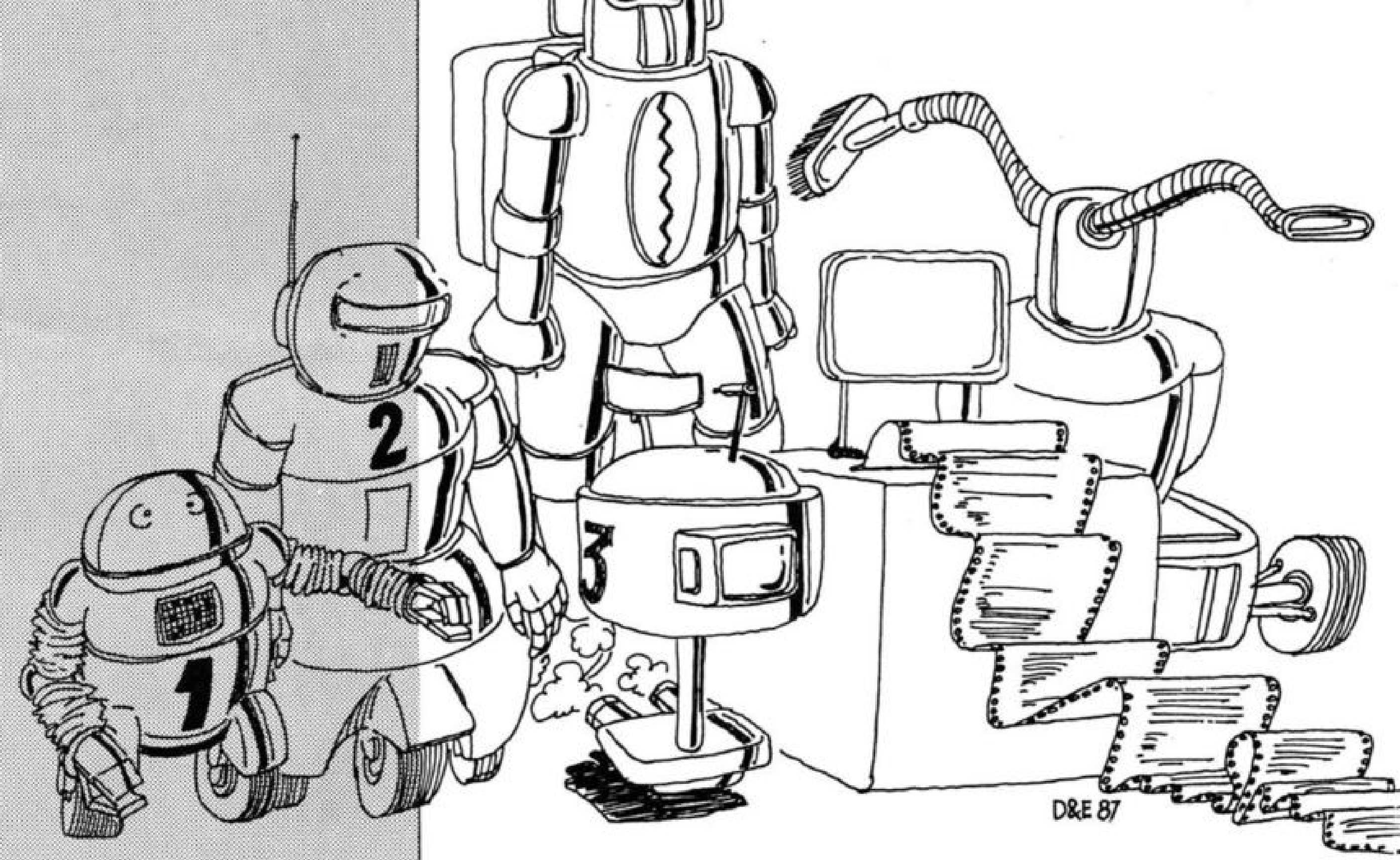
ntecla número de orden de la tecla, de 0 a 5 contadas de izquierda a derecha en la tabla.

En el pequeño ejemplo que se acompaña aparecen las 8 direcciones de las semifilas con los valores de las teclas que están pulsadas. Observa como cambian al pulsar las teclas de tu Spectrum.

DIRECCION:		TECLAS:
65278	-------->	CS a V
65022	-------->	A a G
64510	-------->	Q a T
63486	-------->	1 a 5
61438	-------->	0 a 6
57342	-------->	P a Y
49150	-------->	ENTER a H
32766	-------->	SP a B

```
10 REM **********************
20 REM * LEE TECLADO CON IN *
30 REM **********************
40 REM
50 FOR N=0 TO 7
60 LET A=254+256*(255-2↑N)
70 PRINT AT N,0;A;" --> ";IN A
80 NEXT N
90 GO TO 50
```

LA ORGANIZACION DE LA MEMORIA



LA ORGANIZACION DE LA MEMORIA EN TU SPECTRUM

De todos es sabido, que el Spectrum incorpora como microprocesador, el Z80A, una variación más rápida del Z80 construido en California por Zilog.

El Z8OA es un microprocesador muy rápido, es capaz de procesar 875.000 instrucciones simples por segundo. Además, es capaz de direccionar 65536 direcciones de memoria (64 Kbytes). El Spectrum 48k dispone de las 65536 direcciones, mientras que el 16K solo dispone de 32.768.

ROM	R A M

Figura 1

Estas direcciones de memoria se reparten como sigue:

La ROM (memoria solo de lectura). Ocupa los 16K primeros, y contiene el sistema operativo (7K), el traductor BASIC (8K) y el juego de caracteres (1K). Los integrados que albergan la ROM, estan sellados de forma que no se puede alterar su contenido.

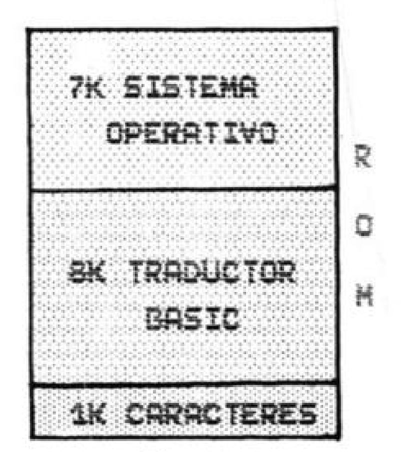


Figura 2

La RAM (Memoria de lectura y escritura). Ocupa en el Spectrum 16K, 16K y en el de 48K, 48K (de ahí su denominación). La información que contienen los chips de la RAM puede modificarse, puesto que estos no están sellados. Con la pérdida de la tensión, los datos de la RAM se borran, por ello es necesario grabarlos en un soporte magnético (disco, cinta ó microdrive).

La memoria RAM se divide en muchas partes, unas de ellas ocupan direcciones fijas, otras varian de tamaño y localización, de forma que para saber donde comienzan, es necesario consultar con la función PEEK las variables del sistema. En la figura 3 se listan los bloques de la RAM con la dirección fija, o con la variable del sistema a leer.

MAPA DE MEMORIA DEL SPECTRUM

0	ROM
16384	ARCHIVO DE PANTALLA
22528	ARCHIVO DE ATRIBUTOS
23296	MEMORIA IMPRESORA
23552	VARIABLES DEL SISTEMA
23734	MAPAS MICRODRIVE
CHANS	INFORMACION CANALES
PROG	PROGRAMA BASIC
VARS	AREA DE VARIABLES
E-LINE	AREA DE EDITAR
WORKSP	AREA DE TRABAJO
STKBOT	PILA DE CALCULADOR
STKEND	MEMORIA DE RESERVA
SP	PILA DE MAQUINA
UDG	GRAFICOS USUARIO

Figura 3

Pero comentemos más detenidamente estas secciones:

• El archivo de pantalla.— Ocupa las direcciones 16384-22527 (6K). A su vez se divide en 3 bloques de 2K cada uno que abarcan, tercios de la pantalla.

A nivel de carácter la pantalla tiene 24 filas × 32 columnas. Normalmente, en BASIC las dos últimas líneas son reservadas para entrada de datos y mensajes de error, por ello no son accesibles con PRINT.

A nivel pixel, la pantalla puede direccionar 256 × 192 puntos, pero por la misma razón que antes, PLOT solo llega hasta 176 en vez de 192.

• El archivo de atributos.— Direcciones 22528-23295 (768 bytes). Ocupa 24 filas × 32 columnas = 768 byte. Se emplea un byte por carácter para definir sus atributos.

La forma en que se calcula es:

Tinta + Papel * 8 + Brillo * 64 + Flash * 128

Cuando usamos la función ATTR (x,y) en BASIC, ésta lo que hace es devolver el valor de la memoria que define los atributos del carácter situado en las coordenadas x e y. Equivale a hacer:

LET ATTR = PEEK (22528 + X * 32 + Y)

• La memoria intermedia de la impresora.— Ocupa las direcciones 23296-23551 (256 bytes). A esta zona se la conoce más comunmente como PRINTER BUFFER. Si el Spectrum no tiene conectada una impresora, esta zona puede ser utilizada para guardar datos, teniendo cuidado de no usar LPRINT, LLIST o COPY.

• Las variables del sistema.— Localizadas en las direcciones 23552-27333 (182 bytes), son las que usa el microprocesador para su trabajo con el sistema.

Las variables pueden ser de 1, 2 o más bytes. Cuando son de 2 bytes se almacena primero el byte "low" (de menor peso) y a continuación el byte "ligh" (de mayor peso). Ejemplo:

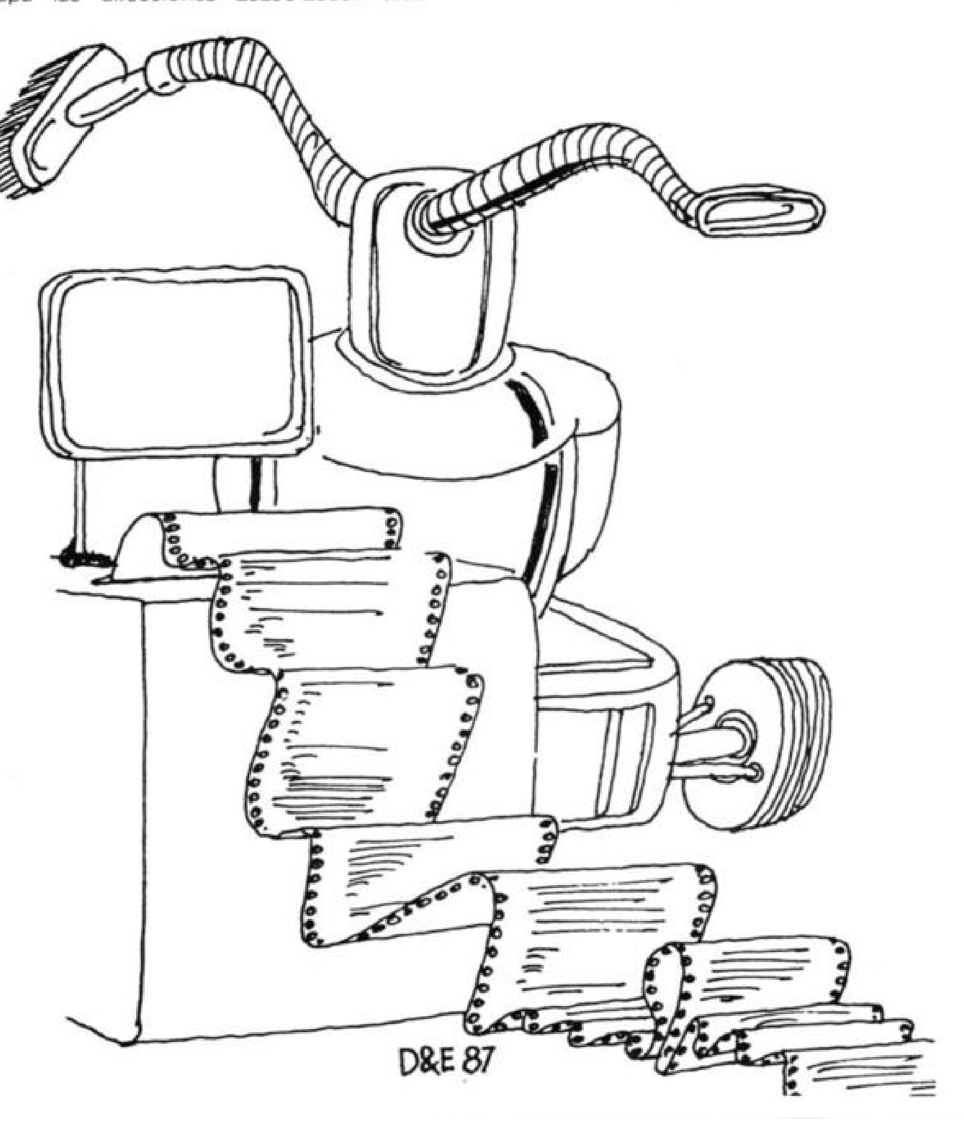
—Para saber donde comienza nuestro programa BASIC haríamos:

LET BASIC = PEEK (23635) + 256 * PEEK (23636)

De estas variables se puede sacar una valiosa información. Además podemos alterar su contenido (con conocimiento de causa) para forzar distintas situaciones. Ejemplo:

POKE 23658,5 obliga a cursor en mayúsculas.

POKE 23658,0 obliga a cursor en minúsculas



• Mapas de Microdrive.— Comienza en la 23734 y finaliza en CHANS-1. Al conectar los microdrives, todo lo que haya por encima de los mapas de microdrives se desplazará hacia arriba tantos bytes como estos ocupan. Por esto, casi todas las zonas que siguen a este guardan su dirección de comienzo en las variables del sistema.

• Información para canales.— Comienza en CHANS (23631 y 23632). Son 4 los canales: K, S, R, P.

• El área del BASIC.— Comienza en PROG (23635 - 23636). A partir de aquí, se almacenan las líneas del programa BASIC:

—Primero el nº de línea. Cambiando el orden normalmente seguido, primero el byte de peso y luego el de menos peso.

—En segundo lugar se almacena la longitud de la línea en bytes.

—La presencia del código 13 nos marca el fin de la línea.

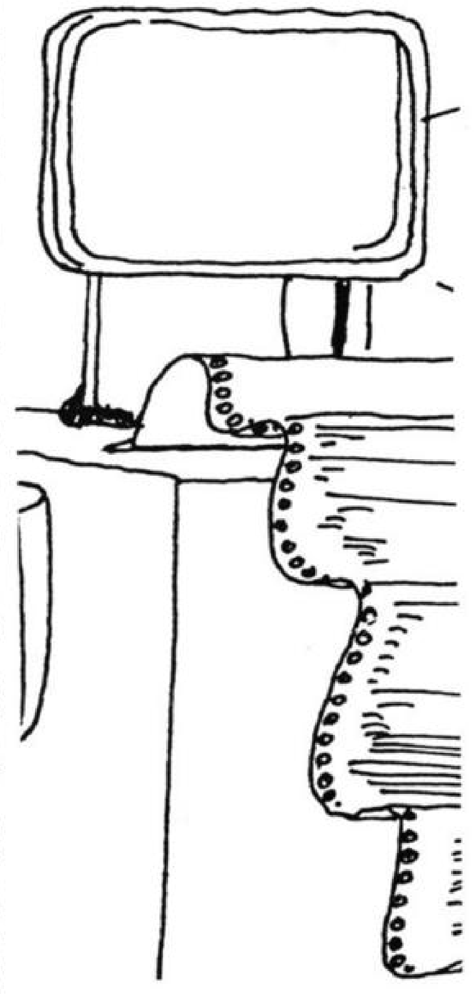
—A continuación vendría la siguiente línea, que en caso de no existir, los códigos 225, 0 marcarán el fin del programa.

• La zona de las variables.— Comienza en VARS (23627 y 23628). Aquí se almacenan las variables, ya sean numéricas o alfanuméricas, o tablas de ambos tipos, la forma en que se almacenan es la siguiente:

—Variable numérica: Código de la letra del nombre, en minúscula (1 byte) y 5 bytes que guardan el valor del número en coma flotante.

—Variable alfanumérica: Código de la letra del nombre en mayúscula (1 byte), 2 bytes con la longitud de la cadena (sólo limitada por la memoria) y los códigos de los caracteres que forman la cadena.

—Tabla alfanumérica (DIM): Se reserva en memoria el número de bytes, resultado del producto de la altura por la anchura.



—Tabla numérica: Igual que en las alfanuméricas, pero reservando 5 bytes para cada elemento.

• El área de editar.— Comienza en la variable E-LINE (23641, 23642), y contiene la línea que se va a editar, o la línea sobre la que estamos trabajando actualmente.

• El área de trabajo.— Comienza en WORKSP (23649 y 23650). Cada vez que se emplea se expande, y cuando deja de ser necesaria, se "limpia" para dejar libres las direcciones de memoria que ocupa.

• La pila del calculador.— Comienza en STKBOT (23651 y 23652) y finaliza en STKEND (23653 y 23654). Su funcionamiento es similar a la pila de máquina, el último en entrar es el primero en salir. En esta zona se guardan en coma flotantes los números que van a ser usados por el calculador.

• La memoria de reserva.— Comienza en STKEND y llega hasta la dirección donde comienza la pila de máquina, su longitud depende de la versión del ordenador.

• La pila de máquina.— Es la que usa el microprocesador para almacenar nº de 2 bytes con el uso de las instrucciones PUSH y POP.

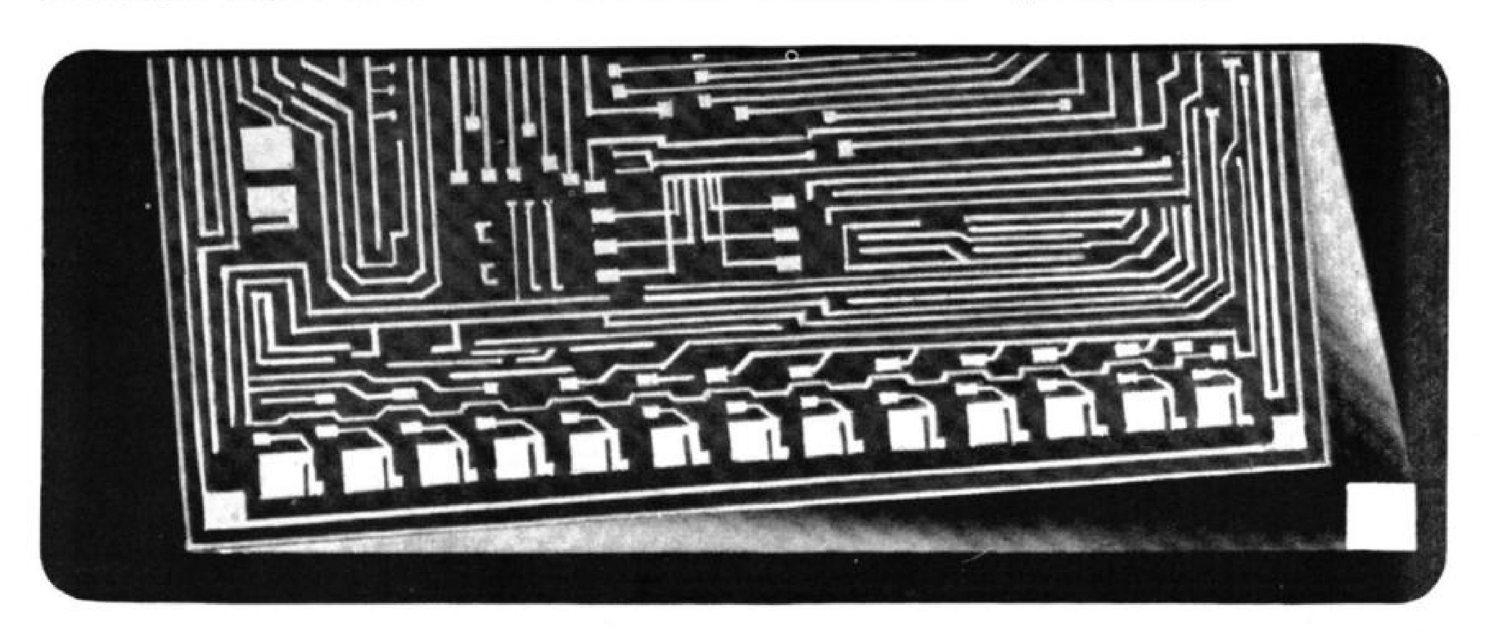
• La pila de GOSUB.— En esta zona, se almacenan los saltos que se realizan en BASIC mediante la instrucción GOSUB, para que al volver con el RETURN, el programa continúe en su sitio.

• La zona de los U.D.G.S. (gráficos definibles por el usuário).— Permite, la redefinición de 21 gráficos. Comienza en UDG (23675, 23676) y tiene una longitud de 168 bytes = 21 × 8.

Es posible cambiar en la variable del sistema UDG, la dirección de comienzo de esta zona, para disponer así de más bancos de gráficos.

El próximo mes volveremos con otro tema de interés que nos permita profundizar en el conocimiento de nuestro Spectrum.

¡¡¡Buen aprendizaje!!!



EL DISCIPLE



El DISCIPLE es un interface de reciente aparición en el mercado, que facilita la conexión directa y simultánea de un Spectrum (sea cual sea el modelo) a:

— Uno o dos joysticks standard

— Una o dos unidades de disco que pueden ser de 5 1/4 ", 3 1/2" ó 3", de 400 de 80 pistas, simple o doble cara y simple o doble densidad. Conexión standard tipo: Shugart SA400.

— Una impresora con conexión paralela Centronics.

Además dispone de:

— Un conector para los restantes periféricos.

— Red local multiusuario, compatible con la Zx Net del Interface 1 de Sinclair, utilizable por 64 usuarios.

— Transfer incorporado: un pulsador permite almacenar en disco el contenido íntegro de la memoria.

Este completo interface se acompaña con un cassette que contiene el programa UTILITIES PROGRAM, que nos permite adaptar cuestiones sobre el número de unidades de disco (1 ó 2), el tipo (40 ó 80 pistas y simple o doble cara) y el step-rate (retardo por paso) del disco. Nos pregunta también si vamos a emplear la impresora, en caso afirmativo no per mite configurarlo, si deseamos que parpadee el borde durante las operaciones en disco, y si se va a usar la red local. Si se va a usar nos pregunta si es la maestra, y si no, el número de estación. También nos permite formatear discos en simple o doble densidad, con o sin sistema.

Por último nos ofrece las opciones de repetir, para corregir errores cometidos en la introducción de datos, y grabar el programa utilities en disco.

"EL DISCIPLE OFRECE TODO UN MUNDO DE POSIBILIDADES"

LOS JOYSTICKS

Para conectar los joysticks disponemos de dos tomas tipo ATARI. Kempstom en la derecha, y Sinclair para la izquierda y derecha.

LAS UNIDADES DE DISCO

Para empezar a trabajar con ellas es necesario cargar el sistema que hemos configurado con el UTILITIES PROGRAM. La forma de hacerlo es pulsar RUN y ENTER tras haber inicializado el Spectrum.

Para seleccionar la unidad escribiremos: D1 para la unidad de disco A, D2 para la B o D * para la actual. Se puede emplear también la sintaxis del microdrive, donde D1 es reemplazado por *"m";1;.

Podemos obtener dos tipos de directorio, uno normal con CAT 1 o CAT 2, y otro simplificado con CAT 1! o CAT 2!.

La información proporcionada en el primer caso es:

El número de orden del programa, el nombre (10 caracteres como máximo), los Kbytes que ocupa, y el distintivo de tipo.

En cambio en el segundo caso solo se listan los nombres.

Las operaciones que se pueden realizar en el disco son:

Grabación, con SAVE D1 "Nombre", que admite autoejecución con LINE xxxx.

Verificación, con VERIFY D1 "Nombre"

Carga, con LOAD D1 "Nombre"

Renombrar un fichero: ERASE D1 "ejemplo 1" TO "ejemplo 2"

Copiar ficheros: COPY D1 "ejemplo 1" TO "copia". Los ficheros Snapshot de los que hablaremos más adelante no pueden ser copiados así. Las copias se pueden realizar dentro del mismo disco, o de un disco a otro.

Borrar ficheros: ERASE D1 "fichero 1"

Crear ficheros autoejecutables: ERASE D1 "fichero" TO "Autoload"

Formatear un disco: FORMAT D1 para hacerlo con doble densidad. FORMAT SD1 para hacerlo en simple densidad.

Esta operación necesita unos 8000 bytes a partir de la dirección 49.152.

Para copiar el sistema:

SAVE D1 "System" CODE 0, 6144.

"LA RED COMUNICA HASTA 64 SPECTRUM"

Salvar ficheros con el botón Snapshot: Es posible grabar un programa en el pto. actual en que se encuentra, pulsando este botón. Sólo se pueden grabar en el DISCO 1, y han de ser programas de 48 K (no son posibles los de 128 K). Los ficheros se grabarán como SNAP A el primero, SNAP B el segundo, etc.

Para cargar un fichero SNAP se debe teclear S después del nombre del fichero. Igualmente se hará si se usa la sintaxis de microdrive.

La pulsación del BREAK interrumpe la ejecución del programa BASIC, pero las operaciones en disco continúan hasta completarse.

Se dispone de una sintaxis abreviada. LOAD P1, carga el programa numerado como 1 en el directorio. VERIFY Y MERGE también se pueden utilizar.

Tenemos además la posiblidad de usar símbolos comodines para COPY, CAT y ERASE.

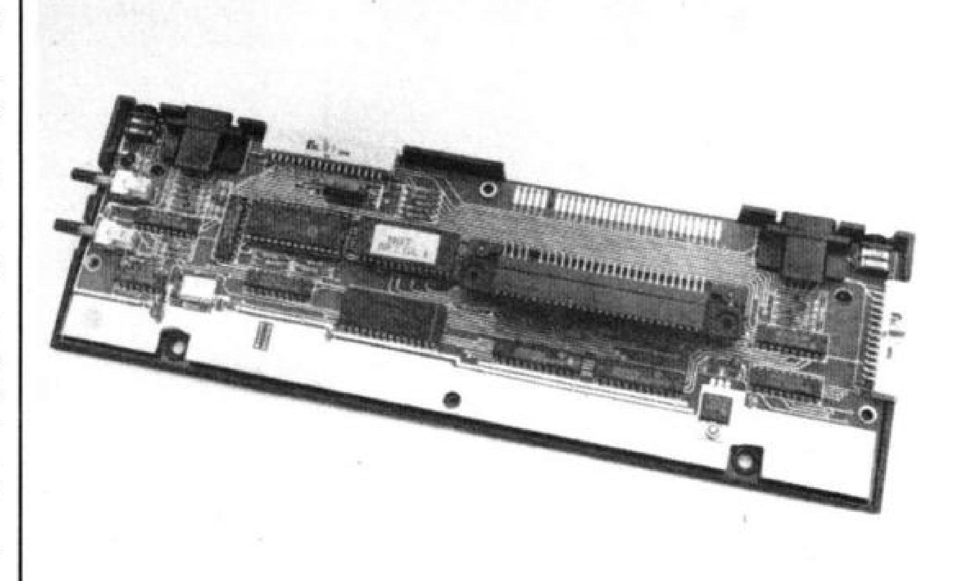
Al introducir un asterisco (*), da igual el resto del nombre. Por ejemplo, ERASE D1 "n*" borra el directorio todos los programas cuyo nombre empiece por "n".

La interrogación (?) es igual que el asterisco, pero para el carácter donde se encuentra. Ejemplo:

ERASE D1 "?YO?TU*", borrará los programas donde sea "YO" la segunda y tercera letra, y "TU" la quinta y sexta.

Si reseteamos nuestro spectrum, para restablecer el sistema operativo habrá que teclear: OUT 123,0.

Junto al botón Snapshot, el interface DISCIPLE lleva el botón de inhibición. Se utiliza si otro periférico no es compatible con el DISCIPLE. Con el botón en su posición más in-



terior e introduciendo OUT 31,0, se desactivan las unidades de disco y la ROM. La salida de impresora y las entradas de los joysticks quedan activas.

Para volver a activar la ROM del DISCIPLE habrá que pulsar de nuevo el botón y teclear OUT 31,16.



COMO USAR LA IMPRESORA

La impresora debe tener la conexión paralela CENTRONIC.

Los comandos que se pueden utilizar son:

LPRINT: Que admite los comandos AT y TAB

LLIST: Lista el programa por impresora.

" SNAPSHOT VUELCA PROGRAMAS A DISCO Y PANTALLAS A IMPRESORA "

COPYSCREEN$: Copia la imagen de la pantalla por impresora.

Para enviar caracteres de control a la impresora, es necesario precederlo del CHR$ (27), uno para cada código a enviar.

Es posible imprimir con el botón Snaspshot. Si se pulsa CAPS SHIFT y manteniéndolo pulsado se pulsa el botón Snaspshot, la pantalla es volcada a la impresora.

LA RED DE RECURSOS COM— PARTIDOS

Es posible unir hasta 63 ordenadores para lo cual se necesita un cable apantallado con un jack standar de 3,5 mm en cada extremo, formando redes de 3 tipos:

—Red de recursos compartidos: Existe una estación Master con unidad de disco e impresora, y 62 posibles estaciones esclavo que se numeran con FORMAT n, donde n es un número de 2-63. Operaciones posibles:

- Master envia fichero a todas las estaciones:

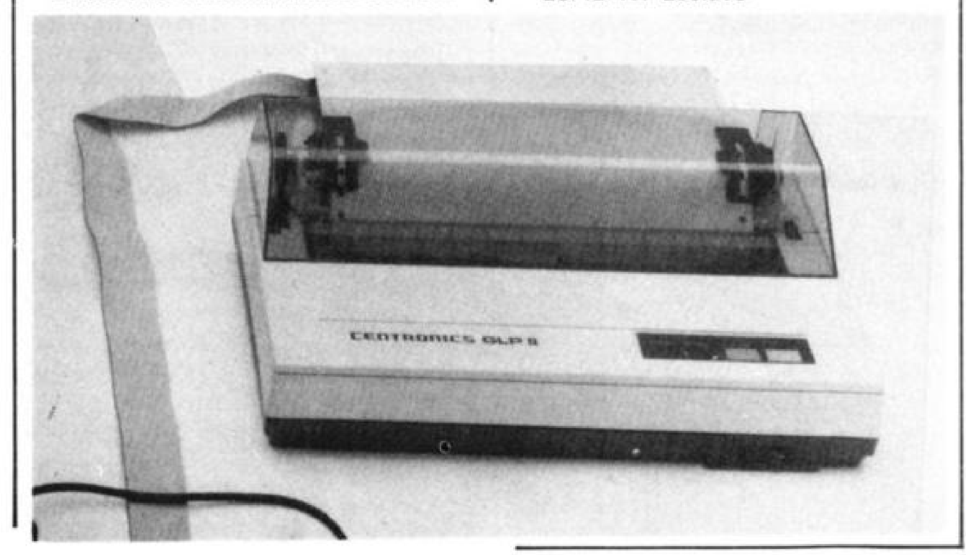
  SAVE NO a la master
  LOAD NO a la esclavo

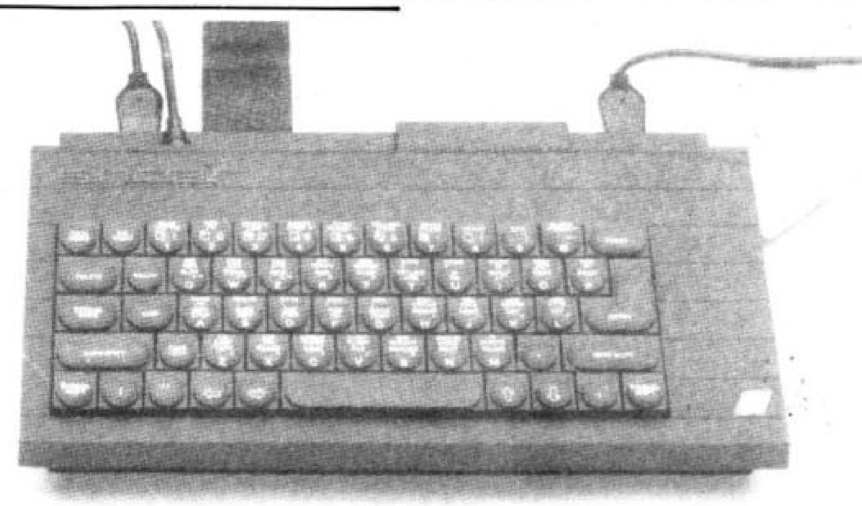
- Esclavo envia fichero a todas las estaciones:

  SAVE NO a la esclavo
  LOAD NO a el resto

- Master envia fichero a una estación:

  SAVE N6 Master
  LOAD N1 Esclavo





• Un esclavo envia un fichero a una estación:

SAVE N6 Master

LOAD N4 estación receptora

• El master inspecciona una pantalla: LOAD N4 Screen$. Se detiene el trabajo del esclavo hasta que la pantalla es transferida a la unidad Master.

• El esclavo puede consultar el directorio del disco Master (CAT D1), cargar un fichero del disco Master (LOAD D1 "Nombre"), enviando (SAVE D1 "Nombre".0) a la impresora. Si ésta está ocupada, el esclavo ha de esperar a que termine la tarea actual antes de continuar. Si varios usuarios acceden a la vez tiene preferencia el de menor número de estación.

RED DE ESTACIONES INDEPENDIENTES

Cada estación dispone de unidad de disco e impresora, y debe configurarse como unidad Master. Para numerarse se emplea FORMAT N con el número de estación (FORMAT N3). Operaciones posibles:

—enviar ficheros a todas las estaciones.—

El emisor SAVE NO

El receptor LOAD NO

—enviar fichero a una sola estación.—

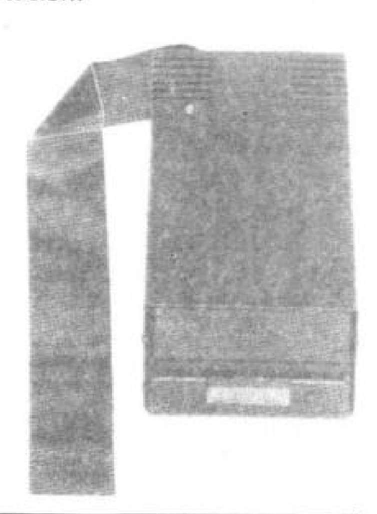
El emisor manda núm. estación receptora

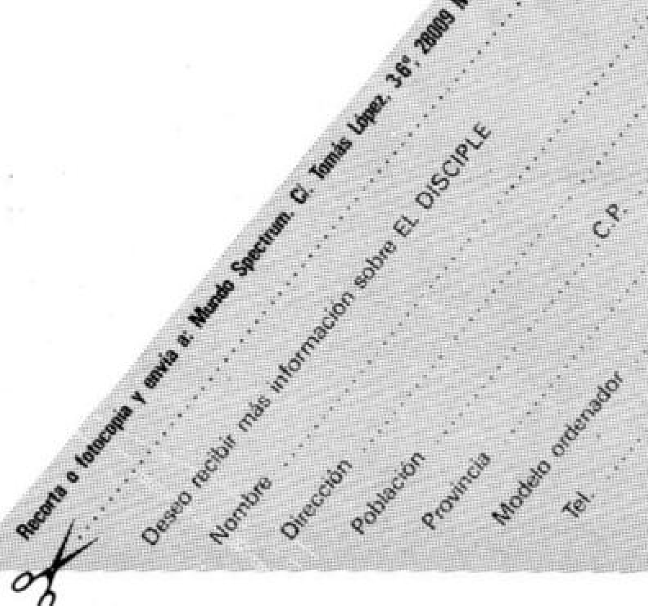
SAVE N4

El receptor identifica estación emisora y teclea LOAD N3

REDES MIXTAS:

Es una red de recursos compartidos con varias estaciones Master.








```
  10 FOR a=0 TO 127: READ b: POK
E USR "a"+a,b: NEXT a
  20 POKE 23658,0
  25 BORDER 0: PAPER 0: INK 4: C
LS
  30 PRINT AT 5,0;"
 NNN
N N  N    N  NNNN          NN   NNN
 NNN      N  N            N  N  N
 N        N  NNN          NNNN  NNN
 N        N  N            N  N  N
 N       NNN NNNN         N  N  N     "
  40 PRINT AT 12,0;"
NNNN  NNN  NNNN  N  N     NNNN  N
N  N   N   N  N  NN N     N     N
N  N   N   N  N  NNNN     N NN  N
NNNN   N   N  N  N NN     N  N  N
       N   NNNN  N  N     NNNN  NNN  "
  50 FOR a=0 TO 28: PRINT INK 6;
AT 20,a;" IJ";AT 21,a;" KL"; INK
5;AT 0,28-a;"IJ ";AT 1,28-a;"KL
"
  52 IF INKEY$<>"" THEN GO TO 90
  54 IF a=28 THEN PRINT AT 20,a;
"  ";AT 21,a;"  ";AT 0,0;"  ";
AT 1,0;"  "
  56 BEEP .005,5: BEEP .05,10: N
EXT a
  58 GO TO 50
  80 POKE 23658,0
  90 BORDER 5: PAPER 7: INK 0: C
LS
 100 LET c=1: LET x=10: LET y=14
: LET xb=x: LET yb=y
 110 LET i=1: LET e=1: LET f=1:
LET eb=e: LET fb=f
 115 LET v=1: LET w=12: LET vb=v
: LET wb=w
 120 LET P=0: LET T=500
 130 LET g=0: LET h=0: LET ai=1:
LET fi=1
```

En este programa apareces como el monstruo de las manzanas, en el que tu voraz apetito te obliga a comer constantemente una manzana tras otra.

Pero mala suerte la tuya, porque el único sitio donde aún quedan manzanas comestibles es dentro del cementerio de la mansión de los Fantasmas sin alma.

Deberás tener cuidado de no chocar contra las lápidas que se encuentren en tu paso, pero más cuidado tendrás que tener aún con el fantasma sin alma



```
200 PRINT PAPER 6; INK 2;AT 0,0;"0000000000000000000000000000000000";AT 19,0;"0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
205 PRINT PAPER 1; INK 7;AT 20,2;" PUNTOS:  ";AT 20,17;" TIEMPO: 500 "
210 FOR a=1 TO 18: PRINT PAPER 6; INK 2;AT a,0;"D";AT a,31;"D": NEXT a
220 IF i=1 THEN FOR a=1 TO 10: PRINT INK 3;AT 1+INT (RND*15),1+INT (RND*29);"M": NEXT a
230 IF i=1 THEN FOR a=1 TO 5: PRINT INK 4;AT 1+INT (RND*16),1+INT (RND*29);"N": NEXT a
1000 REM ********************
1001 REM ** IMPRESION GLOTON **
1002 REM ********************
1005 LET i=3-c
1010 PRINT AT xb,yb;"  ";AT xb+1,yb;"  "
1015 IF i=1 THEN PRINT AT x,y;"AB";AT x+1,y;"CD"
1020 IF i=2 THEN PRINT AT x,y;"EF";AT x+1,y;"GH"
1030 LET xb=x: LET yb=y
1040 LET a$=INKEY$
1050 IF a$="p" THEN LET y=y+1: IF y>29 THEN LET y=29: BEEP .005,30
1060 IF a$="o" THEN LET y=y-1: IF y<1 THEN LET y=1: BEEP .005,30
1070 IF i=1 THEN IF a$="q" THEN LET x=x-1: IF x<1 THEN LET x=1: BEEP .005,30
1080 IF i=1 THEN IF a$="a" THEN LET x=x+1: IF x>17 THEN LET x=17: BEEP .005,30
1100 REM ********************
1101 REM ** IMPRESION MALO **
1102 REM ********************
1110 PRINT INK 7;AT eb,fb;"  ";AT eb+1,fb;"  "
1120 PRINT INK 3;AT e,f;"IJ";AT e+1,f;"KL"
1130 LET eb=e: LET fb=f
1135 IF i=2 THEN GO TO 1200
1140 IF e<x THEN LET e=e+.5
1145 IF e>x THEN LET e=e-.5
1150 IF f<y THEN LET f=f+.5
1155 IF f>y THEN LET f=f-.5
1180 IF i=1 THEN GO TO 1300
1200 LET e=e+ei: LET f=f+fi
1210 IF e>14 THEN LET ei=-2
1220 IF e<10 THEN LET ei=2
1230 IF f>26 THEN LET fi=-2
1240 IF f<4 THEN LET fi=2
1300 GO SUB 2500
1400 REM ********************
1401 REM * IMPR. TIEMPO-PUNTOS *
1402 REM ********************
1410 LET T=T-2
1430 IF T=8 THEN PRINT PAPER 1;AT 20,27;"  "
1440 IF T=98 THEN PRINT PAPER 1; INK 7;AT 20,28;"  "; FLASH 1;AT 20,18;"TIEMPO"; FLASH 0
```

que te perseguirá por todo el cementerio.

Para poder moverte utiliza las teclas siguientes:

q.- arriba, a.- abajo, p.- derecha, o.- izquierda.

Comentario Programa

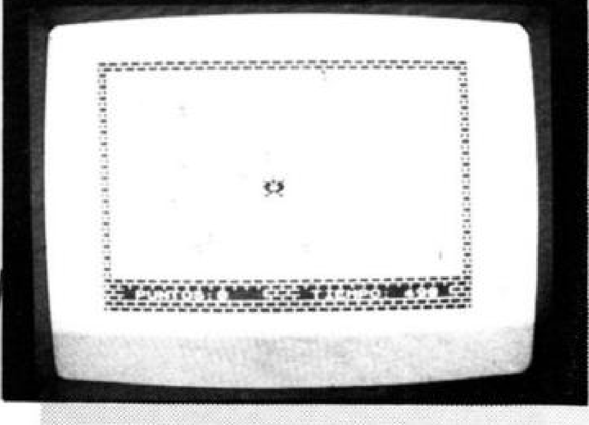
En la línea 10 junto con las líneas que van de la 8000 a 8070 se encuentran los Datos de los UDGs del juego.

De las líneas 25 a la 58 se realiza un pequeño bucle en el que se imprime una pequeña pantalla de presentación antes de empezar el juego.

En la línea 80 se efectua un Poke con la intención de mantener en minúsculas el cursor del ordenador.

En la línea 90 se realiza un CLS de la pantalla y se colorea border, tinta y papel según la necesidad del juego.

De la línea 100 a la línea 130 se inicializan todas las variables del programa, las cuales se co-



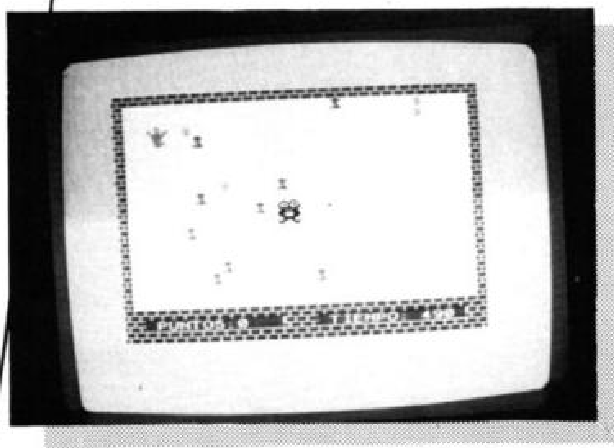
```
1445 IF T<0 THEN LET T=0: GO TO
2600
1447 IF P<0 THEN LET P=0
1450 IF P<100 THEN PRINT PAPER 1
;AT 20,12;" "
1455 IF P<1000 THEN PRINT PAPER
1;AT 20,13;" "
1470 PRINT PAPER 1; INK 7;AT 20,
10;P;AT 20,26;T
1490 IF i=1 THEN GO SUB 1500
1495 IF i=2 THEN GO SUB 1600
1499 GO TO 1000
1500 REM ****************
1501 REM ** IMPR. OBJETOS **
1502 REM ****************
1510 LET g=g+1: LET h=h+1
1520 IF g>15 THEN PRINT INK 4;AT
 1+INT (RND*16),1+INT (RND*25);"
N": LET g=0: BEEP .005,30: BEEP
.009,25
1530 IF h>11 THEN PRINT INK 2;AT
 1+INT (RND*16),1+INT (RND*25);"
M": LET h=0: BEEP .009,5: BEEP .
009,0
1540 RETURN
1597 REM ********************
1598 REM ** MANZANAS COMIDAS **
1599 REM ********************
1600 PRINT AT vb,wb;" "
1610 LET vb=v: LET wb=w
1620 PRINT INK 4;AT v,w;"N"
1625 IF i=2 THEN IF ATTR (x,y)=6
0 OR ATTR (x+1,y)=60 OR ATTR (x+
1,y+1)=60 OR ATTR (x,y+1)=60 THE
N GO TO 2550
1630 IF v>=18 THEN PRINT AT v,w;
"P": FOR a=1 TO 20: BEEP .005,a:
 NEXT a: PRINT AT v,w;" ": LET v
=1: LET w=5+INT (RND*20)
1640 LET v=v+1
1650 RETURN
2500 REM ***************
2501 REM ** COLISIONES **
2502 REM ***************
2510 IF ATTR (x,y)=59 OR ATTR (x
+1,y)=59 OR ATTR (x+1,y+1)=59 OR
 ATTR (x,y+1)=59 THEN GO TO 2600
2520 IF ATTR (x,y)=58 OR ATTR (x
+1,y)=58 OR ATTR (x+1,y+1)=58 OR
 ATTR (x,y+1)=58 THEN LET x=xb:
LET y=yb: FOR a=1 TO 20: OUT 254
,a: NEXT a: BORDER 5: LET P=P-10
: RETURN
2530 IF i=1 THEN IF ATTR (x,y)=6
0 OR ATTR (x+1,y)=60 OR ATTR (x+
1,y+1)=60 OR ATTR (x,y+1)=60 THE
N GO TO 2700
2540 RETURN
2550 FOR a=1 TO 20: BEEP .005,a:
 BORDER INT (RND*6): NEXT a: BOR
DER 5
2551 LET P=P+20
2552 LET v=1: LET w=5+INT (RND*2
0)
2553 RETURN
2600 REM ****************
2601 REM ****  FINAL  ****
2602 REM ****************
```

rresponden de la siguiente manera: **c.**- contador general; **x,y.**- movimiento del traga manzanas; **xb,yb.**- borrado del come manzanas; **i.**- contador parcial; **e,f.**-movimiento fantasma; **eb,fb.**- borrado fantasma; **v,w.**- variables manzana; **vb,wb.**- borrado manzana; **P.**- puntos; **T.**- contador tiempo; **g.**- contador manzanas; **h.**- contador lápidas; **ei,fi.**- variables del movimiento del fantasma, como incremento y decremento.

De la línea 200 a la línea 230 se imprime en pantalla el paisaje o contorno y los objetos.

De la línea 1000 a la línea 1080, aparece el bucle principal y se realiza la impresión y el movimiento de nuestro muñeco.

De la línea 1100 a la línea 1300 se realiza la impresión y el movimiento del fantasma incrementándose y decremen-



```
2608 FOR a=1 TO 70: OUT 254,a: P
RINT INK INT (RND*6);AT x,y;"22"
;AT x+1,y;"22": BEEP .004,INT (R
ND*40): NEXT a: BORDER 5
2610 FOR a=1 TO 18: PRINT INK 5;
 PAPER 1;AT a,1;"000000000000000
0000000000000000": BEEP .005,20:
NEXT a
2615 PRINT PAPER 7; INK 0;AT 5,1
0;"GAME OVER"; PAPER 6;AT 9,10;"
PUNTOS: ";p; PAPER 5;AT 14,8;"OTR
A VEZ (S/N)"
2620 IF INKEY$="s" THEN GO TO 26
28
2622 IF INKEY$="n" THEN STOP
2624 GO TO 2620
2628 FOR a=18 TO 1 STEP -1: PRIN
T INK 0; PAPER 7;AT a,1;"
           ": BEEP .
005,20: NEXT a
2629 LET i=3-i
2630 IF i=1 THEN LET x=12: LET y
=14: LET e=1: LET f=1
2640 IF i=2 THEN LET x=17: LET y
=14: LET e=1: LET f=1
2690 BORDER 5: GO TO 120
2700 FOR a=72 TO 50 STEP -1: OUT
 254,a: BEEP .005,a-50: NEXT a
2710 LET p=p+20
2750 RETURN
5000 STOP
8000 REM ****************
8001 REM *** DATOS UDG: ***
8002 REM ****************
8005 DATA 62,99,205,205,227,127,
61,127,124,198,155,155,199,254,1
88,254
8010 DATA 252,249,124,63,15,22,
248,120,63,159,62,252,240,120,31
,30
8020 DATA 62,127,255,217,227,127
,61,127,124,254,255,179,199,254,
188,254
8030 DATA 251,252,127,62,15,14,8
0,60,223,63,254,252,240,112,56,3
8040 DATA 3,1,3,98,115,93,31,15,
128,192,224,163,239,221,252,248
8050 DATA 15,7,7,7,3,3,1,0,240,2
24,224,216,252,244,191,0
8060 DATA 24,126,126,21,60,44,94
,255,6,8,54,127,93,111,62,28
8070 DATA 0,127,127,127,0,247,24
7,247,4,32,2,144,4,33,3,64
```

A B C D E F G H I J K L M N O P

tándose según se situe nuestro muñeco.

De la línea 1400 a la línea 1499 se realiza la impresión de los puntos y el tiempo en pantalla.

De la línea 1500 a la línea 1540 se realiza la impresión de manzanas y lápidas en pantalla cada cierto tiempo.

De la línea 1597 a la línea 1650 se comprueba cuando nuestro muñeco se come las manzanas.

De la línea 2500 a la línea 2533 se comprueban las colisiones con nuestro tragamanzanas.

De la línea 2600 a la línea 2750 se realiza el final del programa, y se dá la opción de volver a jugar.

Dispones de dos pantallas de animación. SUERTE.

LUNAR—CAR



```
10 REM ************
20 REM * LUNAR-CAR *
30 REM ************
40 REM
50 FOR A=0 TO 167: READ B: POKE USR "A"+A,B: NEXT A
60 LET A=0
70 BORDER 0: PAPER 7: INK 0: CLS
90 REM *******************
91 REM *** DIBUJO PAISAJE ***
92 REM *******************
100 PRINT INK 0;AT 20,0;"ABCABCABCABCABCABCABCABCABCABC..."
110 PRINT INK 0;AT 12,0;"DEFDEFDEFDEFDEFDEFDEFDEFDEFDE"
120 FOR A=0 TO 11: PRINT PAPER 0;AT A,0;"  ": NEXT A
130 PRINT PAPER 0; BRIGHT 1; INK 7;AT 1,0;"..."
135 PRINT PAPER 0; BRIGHT 0; INK 4;AT 3,0;"..."
140 PRINT INK 7; PAPER 1;AT 9,1;"PUNTOS: "; PAPER 2;AT 9,18;"RECORD: "
150 POKE 23658,8
197 REM *****************
198 REM **** VARIABLES ****
199 REM *****************
200 LET X=18: LET XB=X: LET C=1: LET SALTO=0
210 LET DISP=0: LET DX=X: LET DY=9: LET DB=0
220 LET MOVP=1
230 LET ROBOT=0: LET RX=30: LET RXB=RX
```

Después de un viaje interestelar por nuestra Vía Láctea has ido a aterrizar con tu nave nodriza en un planeta desconocido para el planeta tierra.

A bordo de tu LUNAR—CAR provisto de un potente laser, tendrás que hacer frente a robots y numerosos meteoritos que giran alrededor de la órbita del Planeta desconocido Guram.

Intenta acertar con tu laser a la primera sobre tu objetivo o quedarás a expensas del enemigo.

Para moverte por el planeta con el LUNAR—CAR dispones de las siguientes teclas: **Q.- SALTO, P.- DISPARO.**

```
 240 LET P=0: LET CONT=1
 250 LET ROCA=0: LET ROX=15: LET
ROY=30: LET ROXB=ROX: LET ROYB=
ROY
 300 PRINT INK 2;AT X,6;"GHI";AT
X+1,6;"JKL"
 310 PRINT INK 4;AT 14,8;"PULSA
UNA TECLA": PAUSE 30
 320 PAUSE 0
 330 PRINT AT 14,8;"
 "
 997 REM *****************
 998 REM ** IMPRESION CAR **
 999 REM *****************
1000 PRINT AT XB,6;"   ";AT XB+1
,6;"   ": LET XB=X
1010 PRINT INK 2;AT X,6;"GHI";AT
 X+1,6;"JKL"
1100 IF SALTO=0 THEN IF INKEY$="
0" THEN LET SALTO=1
1200 IF SALTO=1 THEN LET X=X-1:
BEEP .003,10: IF X<=13 THEN LET
SALTO=2
1210 IF SALTO=2 THEN LET X=X+1:
BEEP .003,10: IF X>=18 THEN LET
SALTO=0: LET X=18
1300 IF DISP=0 THEN IF INKEY$="P
" THEN LET DISP=1: LET DX=X: LET
 DY=9: LET DB=0: BEEP .05,15
1310 IF DISP=1 THEN GO SUB 2100
1400 IF CONT=1 THEN GO SUB 2500
1410 LET CONT=3-CONT
1500 IF ROBOT=0 AND RND>.9 THEN
LET ROBOT=1: LET RX=30
1510 IF ROBOT=1 THEN GO SUB 3000
1600 IF ROCA=0 AND RND>.9 THEN L
ET ROCA=1: LET ROX=14+INT (RND*5
): LET ROY=30
1610 IF ROCA=1 THEN GO SUB 4000
1700 GO SUB 4500
1800 PRINT INK 7; PAPER 1;AT 9,1
0;P; PAPER 2;AT 9,25;R
2000 GO TO 1000
2097 REM *****************
2098 REM *** DISPARO CAR ***
2099 REM *****************
2100 IF DB=0 THEN PRINT INK 6;AT
 DX,DY;"III"
2105 GO SUB 3500
2110 IF DB=1 THEN PRINT AT DX,DY
;"   "
2115 BEEP .003,8: BEEP .004,12
2120 LET DY=DY+3
2130 IF DY>29 AND DB=0 THEN LET
DB=1: LET DY=9
2140 IF DY>29 AND DB=1 THEN LET
DB=2: LET DY=9: LET DISP=0: RETU
RN
2150 RETURN
2497 REM *****************
2498 REM ** MOVIM. PAISAJE **
2499 REM *****************
2500 IF MOVP=1 THEN PRINT INK 0;
AT 20,0;"BCABCABCABCABCABCABCABC
ABCABCABC";AT 12,0;"EFDEFDEFDEFD
EFDEFDEFDEFDEFDEFDEF"
2510 IF MOVP=2 THEN PRINT INK 0;
AT 20,0;"CABCABCABCABCABCABCABCA
BCABCABCA";AT 12,0;"FDEFDEFDEFDE
FDEFDEFDEFDEFDEFDEFD"
```

Comentario del Programa.

En la línea 50 junto con las DATAS de la 8000 a la 8070, están los datos de los UDGs del juego, los cuales no estarán introducidos en memoria hasta que no se produzca el primer RUN del programa.

En la línea 60 se inicializa la variable del Record.

En la línea 70 se colorea el border, papel y tinta.

Desde la línea 90 a la línea 140 se dibuja el paisaje sobre el cual se va a desarrollar el juego.

En la línea 150 hay un pequeño Poke cuyo objetivo es asegurar que el ordenador va a empezar el juego con el cursor en Mayúsculas.

Desde la línea 197 a la línea 250 se inicializan todas las variables del programa, las cua-

Continúa en pág. 21.



SPIRITS

VIDEO-AVENTURA TOPO SOFT

En esta aventura de Topo Soft, nuestro héroe es un monje, destinado a combatir el espíritu del mal que se encuentra en el interior del castillo.

La forma de conseguirlo es la siguiente:

—Primero hay que ir a la habitación que se nos muestra en la mitad inferior de la pantalla, allí hallaremos la bola mágica que podremos recoger pulsando CAPS SHIFT. La bola nos revelará la posición de los demás objetivos.

—Pulsando ahora la tecla 1 nos aparecerá en la pantalla inferior, el lugar donde se encuentra el libro de hechizos, que tenemos que recoger.

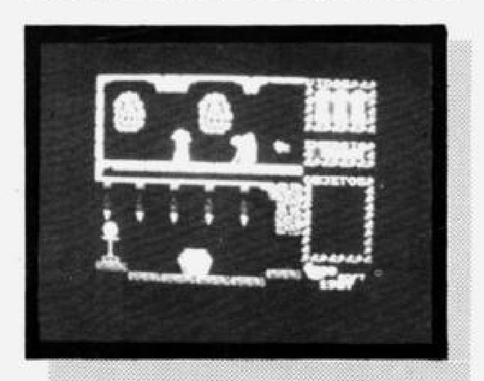
—Pulsando el 2 se hará lo propio indicándonos el lugar donde está la Varita Mágica.

—Una vez tengamos estos dos objetos, pulsando el 3 veremos a un enano vestido de payaso, que al entrar en contacto con nosotros se convertirá en princesa al romperse su encantamiento.

—La princesa nos dará un trozo de piedra, que unido al que nos dejará el caballero medieval al que localizaremos pulsando el 4, formarán el dibujo de un pájaro.

—Y por último, pulsando el 5, podremos localizar el pájaro y tendremos que dispararle para completar nuestra misión.

Pero esta misión se verá dificultada por la presencia de los seres que habitan en el castillo; fantasmas, ogros, etc..., que mediante contacto nos robarán nuestra energía.



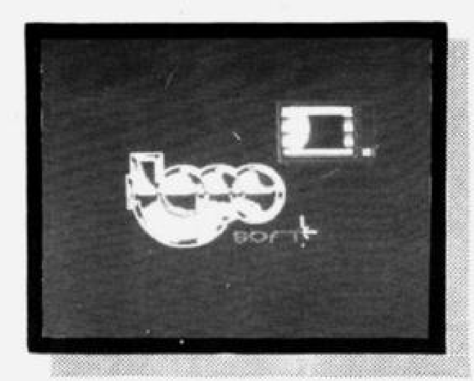
La presentación de este juego ha sido muy cuidada, la carga se realiza de un modo especial y la pantalla de presentación se carga cambiando los colores del borde de una pequeña televisión dibujada en la esquina superior derecha de la misma, y el resto del juego es cargado con una rutina multicolor que finaliza con una melodía.



Los gráficos están muy trabajados y ha habido un buen tratamiento del color. Los movimientos son buenos pero no excesivamente rápidos, de hecho, la presencia de enemigos hace que el juego se haga más lento.

Los sonidos durante el juego son escasos pero cumplen bien su misión.

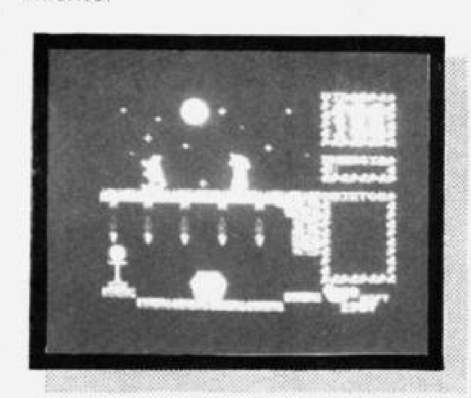
El conocimiento del objetivo a alcanzar da mayor emoción al juego ya que no avanzamos a ver que suerte nos espera sino que hemos de ingeniarnoslas (luchando con nuestro teclado o joystick) para llegar al punto que nos indica la pantalla inferior, librando todos los obstáculos que encontramos en nuestro camino.

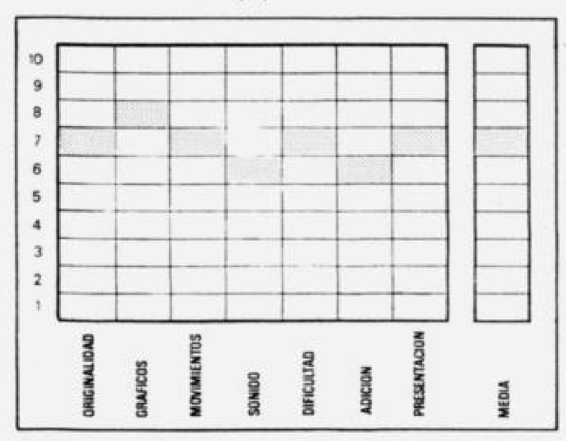


Muchas de las pantallas, poseen puertas ocultas, que podremos abrir activando el correspondiente interruptor, no siempre situado en la misma pantalla que la puerta.

Otra nota curiosa son las lámparas de algunas habitaciones, que en ocasiones nos cortarán el camino cayéndonos encima.

En general se trata de un juego ameno, basado en el típico esquema de pantallas, donde una dificultad no excesiva durante su desarrollo, os proporcionará sin duda largas horas de entretenimiento.







DON QUIJOTE DE LA MANCHA



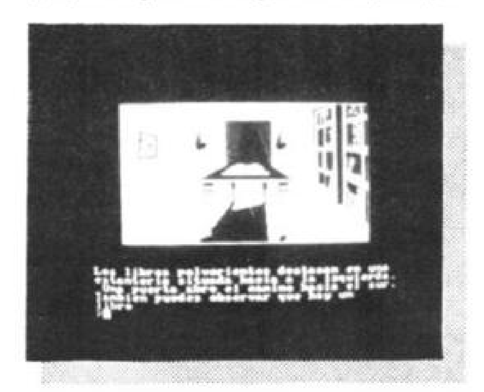
AVENTURA GRAFICO-CONVERSACIONAL DINAMIC

INTRODUCCION:

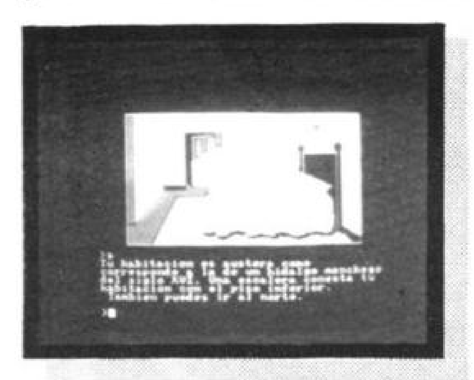
La Mancha, España, abril de 1605...

Eres don Alonso Quijano, un hidalgo manchago del siglo XVI, cuya debilidad son las novelas de caballerías. Todo va más o menos bien por tu casa, hasta que un día tu locura alcanza el punto culminante: decides coger las armas y salir a correr aventuras por la Mancha. ¡Vaya ideas extrañas que tiene el hidalgo! Te consigues un escudero, Sancho Panza, que, aunque no es gran cosa, te ayudará en los momentos difíciles. A partir de este momento entrarás en las más increíbles aventuras que te puedas imaginar. Ventas, colinas, bosques y posadas serán testigos de tus más locas ideas tales como atacar molinos o enfrentarte a fieros leones.

Tu objetivo final es conseguir los amores del Dulcinea del Toboso, tu dama. Aunque existe un primer objetivo, que es armarte caballero, para poder así comenzar tus desventuras.







Comentario:

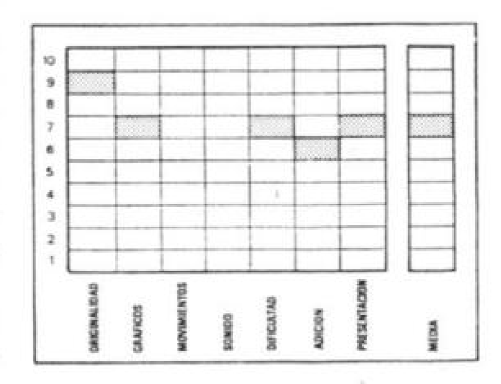
Don Quijote es un juego que pone aprueba tu ingenio y tú capacidad de observación, ya que en él tendrás que resolver muchas situaciones embarazosas, sin más ayuda que la de tú memoria y las pistas que el programa te va dando ocultas o, disfrazadas a veces, entre el texto o diálogo que el ordenador mantiene contigo.

Don Quijote no es en absoluto un juego convencional y tampoco se asemeja a las aventuras grafico-conversacionales clásicas, donde las respuestas del jugador debían ajustarse a un formato rígido y los gráficos eran pocos o de escasa calidad.

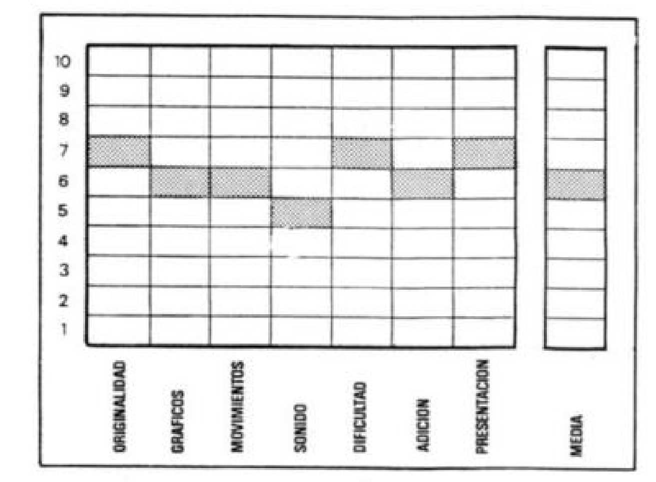
En Don Quijote por el contrario, te encontrarás con que el ordenador comprende frases ligeramente complicadas, y te sorprenderá la gran calidad de los dibujos que aparecen en cada situación. Sin embargo quizá se hechen en falta algunos sonidos ambientales, que hubieran dado mayor emoción al juego.

Como pantalla de presentación tenemos un estupendo dibujo, en el que aparecen Don Quijote y Sancho sobre sus respectivas cabalgaduras, en un paisaje típico manchego. Sin embargo el colorido no es excesivamente bueno.

Como ya hemos dicho, Don Quijote no puede ser comparado con ningún juego convencional ya que su temática es totalmente distinta, sin embargo podemos asegurar sin temor a equivocarnos que es la mejor aventura gráfico-conversacional que se ha realizado en lengua española.



DUSTIN



AVENTURA DINAMIC.

ARGUMENTO:

Año 1989.

La cárcel de alta seguridad de WAD—RAS tiene un preso muy especial, se trata de KID SAGUF, más conocido como DUSTIN, el ladrón de joyas y obras de arte más famoso de su tiempo, que finalmente y tras largas persecuciones de la policía ha sido capturado.

Toda la preocupación de sus guardianes está volcada en envitar que DUSTIN pueda escapar.

Por esa razón todo el centro penitenciario ha sido rediseñado para que nada escape al control de su Director; sin embargo, KID SAGUF ha estado durante meses cavilando, analizando horarios, memorizando el funcionamiento interno del centro y ya tiene su plan para conseguir de nuevo las portadas de todos los periódicos.



Para no olvidar ningún detalle, ha escrito su desarrollo completo en un cuaderno al que hemos tenido acceso y del que aquí no revelamos su información más interesante.

COMENTARIO:

Dustin es un juego que trata un tema bastante original y tiene un alto grado de dificultad, dado el gran número de posibilidades o diferentes situaciones en las que puedes encontrarte, lo cual hace que una vez que has empezado a jugar te sea difícil retirarte de tu ordenador hasta ver si puedes conseguir avanzar un poco más en tu aventura.

El sistema de carga incorpora como ya viene siendo habitual en los programas de SPECTRUM un contador que efectua una cuenta atrás de manera que podemos saber cuanto falta para que empieze el juego. La portada que acompaña la carga es sumamente atractiva y presenta una perspectiva tridimensional que la hace muy realista.

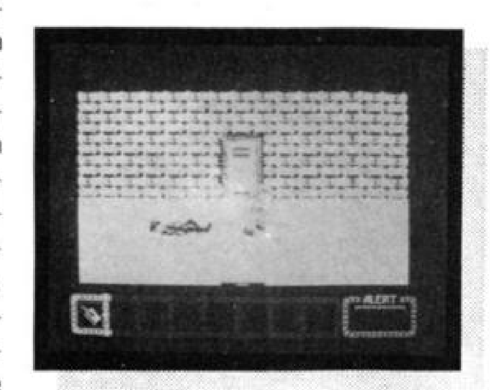
En las celdas y pasillos que constituyen los gráficos del paisaje, se ha realizado también un efecto tridimensional que les da cierta profundidad. Los personajes se mueven con rapidez y Dustin responde inmediatamente a las instrucciones del teclado. En la parte inferior de la pantalla aparecen, en forma de iconos, los diferentes objetos que vamos obteniendo a lo largo del juego, así como un pequeño marcador en la parte



derecha que nos indica el estado de alerta en que nos encontramos.

Los sonidos, si bién existe una bonita melodía durante la presentación, no han sido demasiado trabajados y únicamente nos indican cuándo golpeamos o somos golpeados en nuestros enfrentamientos con los guardias de la cárcel.

Dustin es un juego que como ya hemos dicho presenta un alto grado de dificultad, pero esta dificultad está en avanzar a lo largo de las diferentes etapas de la aventura y no en el manejo del protagonista, que es sumamente sencillo.



```
2520 IF MOVP=3 THEN PRINT INK 0;
AT 20,0;"ABCABCABCABCABCABCABCAB
CABCABCAB";AT 12,0;"DEFDEFDEFDEF
DEFDEFDEFDEFDEFDE"
2530 LET MOVP=MOVP+1: IF MOVP>3
THEN LET MOVP=1
2540 RETURN
2997 REM ******************
2998 REM ** IMPRESION ROBOT **
2999 REM ******************
3000 PRINT AT 18,RXB;"  ";AT 19,
RXB;"  ": LET RXB=RX
3010 PRINT INK 1;AT 18,RX;"EQ";A
T 19,RX;"RS"
3020 LET RX=RX-1
3040 IF RX<0 THEN PRINT AT 18,RX
B;"  ";AT 19,RXB;"  ": LET ROBOT
=0
3050 RETURN
3497 REM ******************
3498 REM *** CHOQUE DISPARO ***
3499 REM ******************
3500 IF ATTR (DX,DY+2)=57 OR ATT
R (DX,DY+3)=57 OR ATTR (DX,DY+4)
=57 OR ATTR (DX,DY+5)=57 THEN GO
 TO 3520
3505 IF ATTR (DX,DY+2)=59 OR ATT
R (DX,DY+3)=59 OR ATTR (DX,DY+4)
=59 OR ATTR (DX,DY+5)=59 THEN GO
 TO 3580
3510 RETURN
3520 LET DY=9: FOR A=1 TO 10: PR
INT AT DX,DY;"  ": PRINT INK IN
T (RND*5);AT 18,RXB;"UU";AT 19,R
XB;"UU": BEEP .004,A: LET DY=DY+
3
3530 IF DY>RX THEN LET DY=9
3540 NEXT A
3545 PRINT AT 18,RXB;"  ";AT 19,
RXB;"  "
3550 LET P=P+20
3560 LET ROBOT=0: LET DISP=0
3570 RETURN
3580 LET DY=9: FOR A=1 TO 10: PR
INT AT DX,DY;"  ": PRINT INK IN
T (RND*5);AT ROXB,ROYB;"UU": BEE
P .004,A: LET DY=DY+3
3590 IF DY>ROY THEN LET DY=9
3600 NEXT A
3610 PRINT AT ROXB,ROYB;"  ": LE
T P=P+100
3620 LET ROCA=0: LET DISP=0
3630 RETURN
3997 REM ******************
3998 REM ** IMPRESION ROCA **
3999 REM ******************
4000 PRINT AT ROXB,ROYB;"  ": LE
T ROXB=ROX: LET ROYB=ROY
4010 PRINT INK 3;AT ROX,ROY;"MN"
4030 LET ROY=ROY-1: IF ROY<0 THE
N PRINT AT ROXB,ROYB;"  ": LET R
OCA=0: RETURN
4040 RETURN
4497 REM ******************
4498 REM ** CHOQUE CONTRA CAR **
4499 REM ******************
4500 IF ATTR (X,9)=57 OR ATTR (X
+1,9)=57 OR ATTR (X,8)=57 OR ATT
R (X+1,8)=57 THEN GO TO 5000
```

les se corresponden de la siguiente manera: **X.**- movimiento del Car, **XB.**- borrado del Car, **C.**- contador general, **SALTO.**- 0 significa que no existe y 1 significa que sí esta saltando, **DISP.**- 0 no hay disparo y 1 se está produciendo el disparo, **DX y DY.**- coordenadas del disparo, **DB.**- borrado disparo, **MOVP.**- movimiento del paisaje, a 1 se mueve a 2 no se mueve, **ROBOT.**- 0 no existe y 1 si exsite, **RX.**- coordenada movimiento robot, **RXB.**- borrado del robot, **P.**- puntos, **CONT.**- contador parcial, **ROCA.**- 0 no existe y 1 si existe, **ROX** y **ROY.**- coordenadas movimiento roca, **ROXB** y **ROYB.**- coordenadas de borrado de la roca.

De la línea 300 a la línea 330 se te dá una opción a una pausa antes de empezar el juego.

De la línea 997 a la línea 2000 se hace la impresión y movimiento del LUNAR-CAR, a la vez que es donde aparecen todas las opciones de disparo y salto de este. (Bucle General).

De la 2097 a la línea 2150 se realiza el bucle de disparo de nuestro CAR.

```
4510 IF ATTR (X,9)=59 OR ATTR (X
+1,9)=59 THEN GO TO 5000
4520 RETURN
4997 REM **************
4998 REM **** FINAL ****
4999 REM **************
5000 FOR A=1 TO 30: PRINT INK 7*
INT (RND*2);AT XB,6;"UUU";AT XB+
1,6;"UUU": BEEP .005,2: OUT 254,
A: BEEP .004,10: NEXT A
5010 BORDER 0
5020 PRINT AT XB,6;"   ";AT XB+1
,6;"   "
5025 IF R<P THEN LET R=P
5030 PRINT INK 7*INT (RND*2);AT
14,10;"GAME OVER";AT 16,8;"OTRA
VEZ (S/N)"
5035 BEEP .05,5: BEEP .005,15: B
EEP .09,10
5040 IF INKEY$="S" THEN CLS : GO
 TO 90
5050 IF INKEY$="N" THEN RANDOMIZ
E USR 0
5060 GO TO 5030
8000 REM *******************
8001 REM *** DATOS GRAFICOS ***
8002 REM *******************
8010 DATA 56,190,222,221,189,189
,219,231,60,123,123,167,223,223,
231,251,206,239,243,251,247,255,
255,255
8020 DATA 231,219,189,189,221,22
2,190,56,251,231,223,223,167,123
,123,60,255,255,255,247,251,243,
239,206
8030 DATA 0,30,63,104,127,64,127
,104,60,126,227,65,198,225,225,1
06,0,124,33,168,252,190,243,239,
8040 DATA 255,225,222,191,55,59,
63,30,255,195,189,126,118,110,12
6,60,254,132,122,252,220,235,252
,120
8050 DATA 3,29,27,59,45,58,15,7,
128,224,203,184,216,240,176,192,
16,24,44,44,54,110,123,255
8060 DATA 1,7,14,31,49,63,28,15,
192,224,116,247,55,255,183,255,1
4,7,31,53,95,127,42,31
8070 DATA 118,227,249,84,254,250
,172,248,0,17,187,238,68,0,0,0,1
96,230,65,24,154,16,196,209
```

A B C D E F G H I J K

L M N O P Q R S T U

De la línea 2497 a la línea 2540 se imprimen distintos suelos y techos para dar la sensación de movimiento del paisaje lunar.

De la línea 2997 a la línea 3050 se realiza la impresión y el movimiento del robot.

De la línea 3497 a la línea 3630 se comprueban todas las colisiones que puedan existir de nuestro disparo.

De la línea 3997 a la línea 4040 se realiza la impresión y el movimiento de la roca lunar.

De la línea 4497 a la línea 4520 se comprueban todas las colisiones que pudieran haber contra nuestro LUNAR-CAR.

De la línea 4997 a la línea 5060 se produce el final del juego, en el cual se imprime en pantalla el clásico Game Over.



SUPER-MUELLE

```
   5 FOR A=0 TO 111: READ B: POK
E USR "a"+A,B: NEXT A
  10 BORDER 0: PAPER 0: INK 7: C
LS
  15 PRINT INK 5;AT 1,0;"
M     M     M     M     M     M"
  20 PRINT INK 6;AT 5,0;"
JK  JK  JKJK  JKJKL  JKJL   JKJK
LL  LL  L  JK LL     LL JK  LL
JK  JK  JKJK  JKJK   JKJK   JKJK
LL  LL  LL    LL     LL JK    LL
JKLJK   JK    JKJKL  JK L   JKJK"
  30 PRINT INK 4;AT 12,0;"
JK  JK  JKJK  JK     JK   JKJK  JK  JK
LL  LL  LL    LL     LL   LL    LLLLL
JK  JK  JKL   JK     JK   JKL   JKLJK
LL  LL  LL    LL     LL   LL    LL  LL
JKLJK   JKJK  JKJK   JKJK JKJK" JK  JK
  32 LET M=2
  34 FOR A=0 TO 29
  36 LET M=3-M: IF M=1 THEN PRIN
T INK 7;AT 19,A;"ABC";AT 20,A;"D
EF"
  38 IF M=2 THEN PRINT INK 7;AT
19,A;"   ";AT 20,A;"GHI"
  40 BEEP .05,10: BEEP .005,12
BEEP .01,15: BEEP .004,20
  41 IF INKEY$<>"" THEN GO TO 50
  42 PRINT AT 19,A;"   ";AT 20,A
;"   "
  44 NEXT A
  48 GO TO 34
  50 BORDER 7: PAPER 7: INK 3: C
LS
  60 LET A=0
  90 REM ***************
  91 REM *** PAISAJE ***
  92 REM ***************
 100 FOR A=0 TO 20 STEP 2: PRINT
INK 1;AT A,0;"JKJKJK";AT A+1,0;
"LJKJKL": NEXT A
 110 FOR A=0 TO 20 STEP 2: PRINT
INK 1;AT A,22;"JKJKJKJKJK";AT A
+1,22;"LJKJKJKJKL": NEXT A
```

De nuevo volvemos a encontrarnos ante un programa cuya dificultad no será difícil de superar.

Se te ha obsequiado con un super muelle, con el cual deberás lanzar hacia arriba la bola de metal que viene rebotando en las paredes. Si hubiera sido una simple raqueta en vez del muelle, no hubieras podido catapultar las bolas metálicas hacia arriba debido a su peso.

A medida que el juego vaya transcurriendo, la dificultad te irá aumentando hasta tal punto, que si consigues llegar sin fallar te encontrarás en pantallas con tres bolas metálicas, que tendrás que ir catapultando hacia arriba sucesivamente, sin poder fallar ninguna pues quedarías destruido.

Para poder mover el Super-Muelle de un lado hacia otro, se han seleccionado las siguientes teclas en la consola de tu ordenador: **P.- derecha, Q.- izquierda.**

```
120 PRINT INK 1;AT 20,6;"JKJKJKJKJKJKJKJK";AT 21,5;"JKJKJKJKJKJKJKJKJK"
130 PRINT INK 1;AT 0,6;"JKJKJKJK";AT 1,5;"JKJKLLJKJK"
140 PRINT INK 3;AT 0,10;"LLLLLLLL";AT 1,10;"LLLLLLLL"
150 PRINT PAPER 6; INK 0;AT 4,24;"PUNTOS";AT 6,24;"      ";PAPER 5;AT 10,24;"RECORD";AT 12,24;"      "; PAPER 2; INK 7;AT 16,24;" BOLA ";AT 18,24;"      "
190 REM ****************
191 REM **** VARIABLES ****
192 REM ****************
200 LET Y=12: LET YB=Y
205 LET A$="ABC": LET B$="DEF"
210 LET X=3: LET Z=13: LET XB=X: LET ZB=Z: LET XI=1: LET ZI=1
220 LET X1=3: LET Z1=13: LET XB1=X1: LET ZB1=Z1: LET XI1=1: LET ZI1=1
225 LET X2=3: LET Z2=13: LET XB2=X2: LET ZB2=Z2: LET XI2=1: LET ZI2=1
230 LET P=0: LET BO=1
240 LET T=0
900 REM ***************
901 REM ** SALIDA BOLA **
902 REM ***************
910 FOR A=17 TO 10 STEP -1: PRINT AT 0,A;" ": BEEP .005,10: BEEP .009,15: BEEP .004,20: NEXT A
912 FOR A=17 TO 10 STEP -1: PRINT AT 1,A;" ": BEEP .005,10: BEEP .009,15: BEEP .004,20: NEXT A
915 BEEP .3,15: BEEP .5,10
917 IF BO=1 THEN LET CO=2
918 IF BO=2 THEN LET CO=4
919 IF BO=3 THEN LET CO=3
920 FOR A=0 TO 3: PRINT INK CO;AT A,13;"M": BEEP .01,20: BEEP .009,15: BEEP .03,25: PRINT AT A,13;" ": NEXT A
930 PRINT INK CO;AT 3,13;"M"
990 PRINT INK 1+INT (RND*5);AT 8,10;"PULSA UNA";AT 10,12;"TECLA": BEEP .003,INT (RND*20)
992 IF INKEY$<>"" THEN PRINT AT 8,10;"         ";AT 10,12;"     ": FOR A=1 TO 10: BEEP .05,A: NEXT A: GO TO 996
993 PRINT INK 0;AT 18,Y;"ABC";AT 19,Y;"DEF"
994 GO TO 990
996 FOR A=10 TO 17: PRINT INK 3;AT 0,A;"L": BEEP .005,10: BEEP .009,15: BEEP .004,20: NEXT A
997 FOR A=10 TO 17: PRINT INK 3;AT 1,A;"L": BEEP .005,10: BEEP .009,15: BEEP .004,20: NEXT A
998 BEEP .5,10
1000 REM ****************
1001 REM ** IMPRES. MUELLE **
1002 REM ****************
1010 PRINT INK 7;AT 18,YB;"   ";AT 19,YB;"   ": LET YB=Y
1020 IF A$<>"ABC" THEN PRINT INK 0;AT 19,Y;B$
```

Comentario del Programa.

En la línea 50 junto con los DATA de las líneas 8000 a la línea 8060, están los datos de los UDGs del juego, que se te introducirán en memoria cuando des el primer RUN del programa.

En la línea 10 se pone de la pantalla después de realizar un CLS el color del border, papel y tinta según lo requieran las circunstancias.

Desde la línea 15 a la línea 48 aparece un For-Next, en el cual se dibuja una pantalla de presentación antes de la iniciación del juego. Dentro del For-Next, en la línea 41 se dá la opción de salir de él al pulsar cualquier tecla del ordenador.

En la línea 50 procedente de un GOTO de la 41, se realiza un CLS de la pantalla del TV y se colorea según lo deseado.



```
1030 IF A$="ABC" THEN PRINT INK
0;AT 18,Y;A$;AT 19,Y;B$
1040 IF X<18 OR X1<18 THEN LET A
$="ABC": LET B$="DEF"
1050 IF INKEY$="P" OR INKEY$="p"
 THEN LET Y=Y+1: IF Y>19 THEN LE
T Y=19
1060 IF INKEY$="Q" OR INKEY$="q"
 THEN LET Y=Y-1: IF Y<6 THEN LET
 Y=6
1100 GO SUB 2000
1110 PRINT PAPER 6; INK 0;AT 6,2
5;P
1120 PRINT PAPER 5; INK 0;AT 12,
25;R
1130 PRINT PAPER 2; INK 7;AT 18,
26;BO
1140 IF T>400 THEN GO TO 1010
1150 IF T=200 THEN FOR A=1 TO 20
: OUT 254,A: NEXT A: BORDER 7: L
ET BO=2: LET T=T+1: GO TO 910
1160 IF T=400 THEN FOR A=1 TO 20
: OUT 254,A: NEXT A: BORDER 7: L
ET BO=3: LET T=T+1: GO TO 910
1170 LET T=T+1
1200 GO TO 1010
1997 REM *********************
1998 REM ** MOVIMIENTO BOLA 1 **
1999 REM *********************
2000 PRINT AT XB,ZB;" ": LET XB=
X: LET ZB=Z
2010 PRINT INK 2;AT X,Z;"M"
2015 IF BO>1 THEN GO SUB 3000
2020 LET X=X+XI: LET Z=Z+ZI
2030 IF Z<=6 THEN LET ZI=1: BEEP
 .005,10
2035 IF Z>=21 THEN LET ZI=-1: BE
EP .005,10
2040 IF X<=2 THEN LET XI=1: BEEP
 .005,10
2045 IF X=18 AND Z>Y-1 AND Z<Y+3
 THEN GO TO 2100
2050 IF X>=20 THEN GO TO 2200
2060 RETURN
2097 REM *********************
2098 REM *** CHOQUE BOLAS ***
2099 REM *********************
2100 LET A$="   ": LET B$="GHI"
2110 BEEP .005,10: BEEP .05,5: B
EEP .008,20
2120 IF X>=17 THEN LET XI=-1
2125 IF X1>=17 THEN LET XI1=-1
2127 IF X2>=17 THEN LET XI2=-1
2130 LET P=P+5
2140 RETURN
2197 REM *********************
2198 REM ** FINAL PROGRAMA **
2199 REM *********************
2200 IF X>=17 THEN PRINT AT XB,Z
B;"N": BEEP .5,5: BEEP .2,10: BE
EP .1,15: PRINT AT XB,ZB;" "
2205 IF X1>=17 THEN PRINT AT XB1
,ZB1;"N": BEEP .5,5: BEEP .2,10:
 BEEP .1,15: PRINT AT XB1,ZB1;"
"
2207 IF X2>=17 THEN PRINT AT XB2
,ZB2;"N": BEEP .5,5: BEEP .2,10:
 BEEP .1,15: PRINT AT XB2,ZB2;"
"
2210 PRINT INK 7; PAPER 2;AT 8,1
2;"GAME";AT 10,12;"OVER"
```

En la línea 60 se inicializa la variable R correspondiente al record.

Desde la línea 90 a la línea 150 se dibuja el paisaje sobre el cual se va a desarrollar el juego, a la vez que se han impreso los puntos, el record, y el número de bolas que existen.

Desde la línea 190 a la línea 240 se inicializan todas las variables existentes en el juego o programa, correspondiendose de la siguiente manera: **Y.**- movimiento lateral del muelle; **YB.**- borrado del muelle; **A$,B$.**- variables para impresión del muelle; **X,Z.**- variables x,y para movimiento de la primera bola; **XB,ZB.**-borrado de la primera bola; **XI,ZI.**- incremento o decremento de la primera bola; **X1,Z1.**- variables x,y para movimiento de la segunda bola; **XB1,ZB1.**- borrado de la segunda bola; **XI1,ZI1.**- incremento o decremento de la segunda bola; **X2,Z2.**- variables x,y para movimiento de la tercera bola ; **XB2,ZB2.**- borrado de la tercera bola; **XI2,ZI2.**- incremento o decremento de la tercera bola; **P.**- puntos; **BO.**- contador de las bolas que hay en pantalla; **T.**- tiempo.

Desde la línea 900 a la línea 994 se produce la apertura de la puerta superior y salida de las bolas correspondientes al transcurso del juego.

Desde la línea 996 a la línea 998 se vuelve a cerrar la puerta superior.

```
2215 PRINT INK 0;AT 18,YB;"ABC";
AT 19,YB;"DEF"
2220 PRINT INK 7; PAPER 1;AT 14,
7;"OTRA VEZ (S/N)"
2225 IF R<P THEN LET R=P
2230 IF INKEY$="S" OR INKEY$="s"
THEN CLS : GO TO 100
2240 IF INKEY$="N" OR INKEY$="n"
THEN STOP
2250 GO TO 2230
2997 REM ********************
2998 REM ** MOVIMIENTO BOLA 2 **
2999 REM ********************
3000 PRINT AT XB1,ZB1;" ": LET X
B1=X1: LET ZB1=Z1
3010 PRINT INK 4;AT X1,Z1;"M"
3012 IF BO>2 THEN GO SUB 3100
3015 LET X1=X1+XI1: LET Z1=Z1+ZI
1
3020 IF Z1<=6 THEN LET ZI1=1: BE
EP .005,10
3030 IF Z1>=21 THEN LET ZI1=-1:
BEEP .005,10
3040 IF X1<=2 THEN LET XI1=1: BE
EP .005,10
3050 IF X1=18 AND Z1>Y-1 AND Z1<
Y+3 THEN GO TO 2100
3060 IF X1>=20 THEN GO TO 2200
3070 RETURN
3097 REM ********************
3098 REM ** MOVIMIENTO BOLA 3 **
3099 REM ********************
3100 PRINT AT XB2,ZB2;" ": LET X
B2=X2: LET ZB2=Z2
3110 PRINT INK 3;AT X2,Z2;"M"
3120 LET X2=X2+XI2: LET Z2=Z2+ZI
2
3130 IF Z2<=6 THEN LET ZI2=1: BE
EP .005,10
3140 IF Z2>=21 THEN LET ZI2=-1:
BEEP .005,10
3150 IF X2<=2 THEN LET XI2=1: BE
EP .005,10
3160 IF X2=18 AND Z2>Y-1 AND Z2<
Y+3 THEN GO TO 2100
3170 IF X2>=20 THEN GO TO 2200
3180 RETURN
7997 REM ********************
7998 REM **** DATOS UDGS ****
7999 REM ********************
8000 DATA 127,255,227,1,3,6,12,1
8,255,255,195,129,195,102,60,24
8010 DATA 254,255,199,128,192,96
,48,48,6,115,249,231,91,61,60,24
8020 DATA 102,195,153,231,219,18
0,60,24,96,206,159,231,218,136,6
0,24
8030 DATA 127,255,195,125,231,21
9,61,60,255,255,195,189,231,219,
189,60
8040 DATA 254,255,195,190,231,21
9,188,60,0,127,239,219,239,159,2
39,127
8050 DATA 0,252,182,90,222,238,1
90,252,0,124,214,190,214,218,238
,124
8060 DATA 60,110,223,223,207,231
,110,60,18,68,128,17,68,34,8,84

A B C D E F G H I J K L M N
```

Desde la línea 1000 a la línea 1200 aparece el bucle principal, donde se imprime el muelle y se te dá la opción de poderlo mover hacia los lados. También se imprimen puntos y récord.

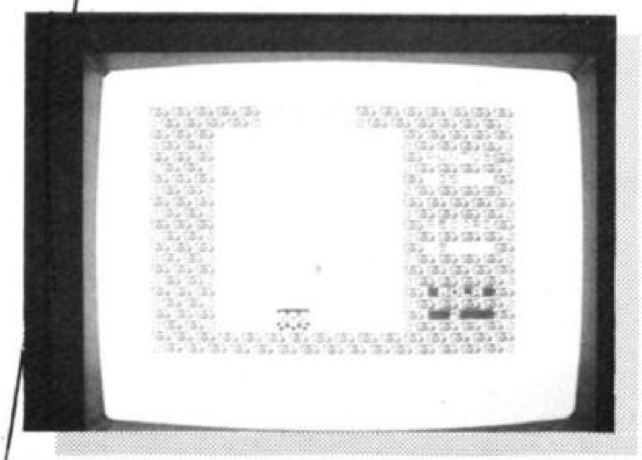
Desde la línea 1997 a la línea 2060 se produce la impresión y el movimiento de la primera bola.

Desde la línea 2097 a la línea 2140 se comprueban los choques de las bolas contra el suelo.

Desde la línea 2197 a la línea 2250 se realiza el final del programa donde explota la bola y se te imprime Game Over al no tener posibilidad de más vidas.

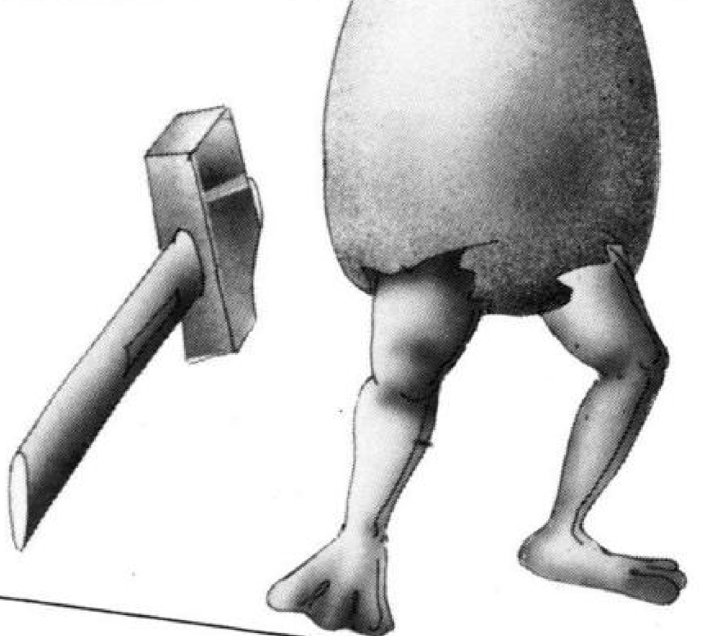
Desde la línea 2997 a la línea 3070 se realiza la impresión y el movimiento de la segunda bola.

Desde la línea 3097 a la línea 3180 se realiza la impresión y el movimiento de la tercera bola.





MISTER–HUEVO



```
1 FOR a=0 TO 167: READ b: POKE USR "a"+a,b: NEXT a
3 BORDER 0: PAPER 0: INK 7: CLS
4 POKE 23658,0
5 PRINT INK 2;AT 5,0;"[illegible]"
6 PRINT INK 6;AT 13,0;"[illegible]"
7 PAUSE 10: PAUSE 0
19 LET r=0
20 BORDER 0: PAPER 0: INK 7: CLS
30 LET a$="GH": LET b$="KL": LET y=15: LET yb=y: LET y1=1: LET y2=1: LET y3=0
40 LET re=0: LET co=1: LET t=500: LET p=0: LET pa=1: LET pae=1
50 LET m=9: LET n=10: LET mb=m: LET nb=n
55 LET g=12: LET h=17: LET gb=g: LET hb=h
60 LET e=14: LET f=23: LET eb=e: LET fb=f
97 REM ***********************
98 REM ** IMPRES. PAISAJE **
99 REM ***********************
100 PRINT INK 2: PAPER 6;AT 0,0;"[illegible]"
```

En este programa tan especial por su contenido, se te anima como a un pollo que al intentar salir del cascarón solamente consigue sacar las patas.

Sin poder ver nada a consecuencia de la cáscara del huevo andas de un lado hacia otro buscando a mamá gallina para que te ayude.

Pero no con mucha suerte, pues te has metido en una fábrica donde se les caen a los obreros martillos, que deberás esquivar para que no te dañen. Pero cuidado con las cintas transportadoras que te harán andar aunque tú no quieras.

Tendrás que moverte de una puerta a otra constantemente, con las siguientes teclas: **P.- derecha, Q.- izquierda.**

```
 110 PRINT INK 2; PAPER 6;AT 19,
0;"SSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSS"
 120 PRINT INK 5;AT 19,5;"ABBBBB
BBBC";AT 19,17;"ABBBBBBBBC"
 125 PRINT AT 20,6;"          ";AT
 20,18;"          "
 130 FOR a=5 TO 15: PRINT INK 2;
 PAPER 6;AT a,0;"SS";AT a,30;"SS
": NEXT a
 140 PRINT INK 2; PAPER 6;AT 5,2
;"SSS";AT 5,27;"SSS";AT 6,2;"SS"
;AT 6,28;"SS";AT 7,2;"S";AT 7,29
;"S";AT 5,14;"SSSSS";AT 6,15;"SS"
 145 PRINT INK 2; PAPER 6;AT 15,
2;"SSS";AT 15,27;"SSS";AT 14,2;"
SS";AT 14,28;"SS";AT 13,2;"S";AT
 13,29;"S"
 150 PRINT INK 7; PAPER 1;AT 1,2
;"PUNTOS:     ";AT 3,2;"RECORD:
    ";AT 1,18;"TIEMPO: 500 ";AT
 3,18;"RECOGIDAS:  "
 160 FOR a=16 TO 18: PRINT FLASH
 1; INK 5; PAPER 1;AT a,0;"■"; F
LASH 0;AT a,31;"■": NEXT a
1000 REM ******************
1001 REM ** IMPRES. HUEVO **
1002 REM ******************
1010 PRINT AT 17,yb;"  ";AT 18,y
b;"  "
1020 LET yb=y
1030 PRINT BRIGHT 1; INK 7;AT 17
,y;a$;AT 18,y;b$; BRIGHT 0
1040 IF INKEY$="p" THEN LET y=y+
1: LET y1=3-y1: LET y3=0: IF y>2
9 THEN LET y=29
1050 IF INKEY$="q" THEN LET y=y-
1: LET y2=3-y2: LET y3=1: IF y<1
 THEN LET y=1
1060 IF y3=0 THEN IF y1=1 THEN L
ET a$="GH": LET b$="KL"
1070 IF y3=0 THEN IF y1=2 THEN L
ET a$="GH": LET b$="IJ"
1080 IF y3=1 THEN IF y2=1 THEN L
ET a$="MN": LET b$="QR"
1090 IF y3=1 THEN IF y2=2 THEN L
ET a$="MN": LET b$="OP"
1100 IF y=1 THEN IF pae=1 THEN G
O SUB 2100
1110 IF y=29 THEN IF pae=0 THEN
GO SUB 2100
1200 REM ******************
1201 REM ** MOVIM. CINTA **
1202 REM ******************
1210 IF co=1 THEN PRINT INK 5;AT
 19,5;"ABBBBBBBBC";AT 19,17;"DEE
EEEEEEF"
1220 IF co=2 THEN PRINT INK 5;AT
 19,5;"DEEEEEEEEF";AT 19,17;"ABB
BBBBBBC"
1230 LET co=3-co
1240 IF y>4 THEN IF y<15 THEN IF
 INKEY$="" THEN LET y=y-1: BEEP
.005,20: BEEP .005,15
1245 IF y>16 THEN IF y<27 THEN I
F INKEY$="" THEN LET y=y+1: BEEP
 .005,20: BEEP .005,15
```

Comentario del Programa.

Empieza en la línea 1 y junto con la líneas que van de la 9000 a la 9060 se introducen mediante Datas los UDGs definidos por el Usuario.

En la línea 3 se empieza en programa coloreando border, papel y tinta, según la necesidad de la pantalla.

En la línea 5 se realiza un Poke, cuyo objeto es mantener en minúscula durante todo el juego el cursor del ordenador.

Desde la línea 5 a la línea 7 te he dibujado una pequeña pantalla de presentación antes de comentar el programa.

En la línea 19 se inicializa la variable r que corresponde al record.

En la línea 20 se realiza un CLS de toda la pantalla del ordenador y se la colorea según la necesidad del juego.

Desde la línea 30 a la línea 60 se inicializan todas las variables que se van a utilizar den-



```
1300 REM *******************
1301 REM ** MOVIM. MARTILLOS **
1302 REM *******************
1310 PRINT AT mb,nb;" "
1320 LET mb=m: LET nb=n
1330 PRINT INK 6;AT m,n;"U"
1380 LET m=m+1: IF m>18 THEN PRI
NT INK 6;AT mb,nb;"I": FOR a=1 T
O 5: BEEP .005,a: NEXT a: PRINT
AT mb,nb;" ": LET m=9: LET n=5+I
NT (RND*9)
1400 PRINT AT eb,fb;" "
1410 LET eb=e: LET fb=f
1420 PRINT INK 3;AT e,f;"U"
1450 LET e=e+1: IF e>18 THEN PRI
NT INK 3;AT eb,fb;"I": FOR a=1 T
O 5: BEEP .005,a: NEXT a: PRINT
AT eb,fb;" ": LET e=9: LET f=17+
INT (RND*9)
1500 PRINT AT gb,hb;" "
1510 LET gb=g: LET hb=h
1520 PRINT INK 4;AT g,h;"U"
1550 LET g=g+1: IF g>18 THEN PRI
NT INK 4;AT gb,hb;"I": FOR a=1 T
O 5: BEEP .005,a: NEXT a: PRINT
AT gb,hb;" ": LET g=11: LET h=6+
INT (RND*20)
1600 IF g=17 OR e=17 OR m=17 THE
N GO SUB 3000
1705 PRINT INK 7; PAPER 1;AT 1,9
;p;AT 3,9;r;AT 1,26;t;AT 3,28;re
1720 LET t=t-1: IF t<0 THEN GO T
O 3500
1730 IF t=99 THEN PRINT PAPER 1;
AT 1,28;" "
1735 IF t=9 THEN PRINT PAPER 1;A
T 1,27;" "
2000 GO TO 1010
2097 REM *******************
2098 REM ** CAMBIO PUERTAS **
2099 REM *******************
2100 FOR a=1 TO 30: OUT 254,a: B
EEP .004,a: NEXT a: BORDER 0
2110 IF pae=1 THEN IF pa=1 THEN
FOR a=16 TO 18: PRINT FLASH 1; I
NK 1; PAPER 5;AT a,0;"■"; FLASH
0;AT a,31;"■": BEEP .009,2: NEXT
a: LET pa=2: LET pae=0: LET p=p
+10: LET re=re+1: RETURN
2120 IF pae=0 THEN IF pa=2 THEN
FOR a=16 TO 18: PRINT FLASH 1; I
NK 1; PAPER 5;AT a,0;"■"; FLASH
0;AT a,31;"■": BEEP .009,2: NEXT
a: LET pa=1: LET pae=1: LET p=p
+10: LET re=re+1
2150 RETURN
3000 REM *******************
3001 REM ** CHOQUE MARTILLOS **
3002 REM *******************
3010 IF m=17 THEN IF n>y-1 AND n
<y+2 THEN GO TO 3500
3020 IF e=17 THEN IF f>y-1 AND f
<y+2 THEN GO TO 3500
3030 IF g=17 THEN IF h>y-1 AND h
<y+2 THEN GO TO 3500
3040 RETURN
3497 REM *******************
3498 REM ** FINAL PROGRAMA **
3499 REM *******************
```

tro del programa y que darán movimiento a los distintos objetos. Estas vienen relacionadas de la siguiente manera: Las variables **a$** y **b$** se las iguala a los gráficos del muñeco, gracias a lo cual, cambiando en el juego éstas, se hará que parezca que el muñeco se mueve andando de un sitio a otro; **y.-** coordenada y del muñeco o huevo; **y1, y2, y3.-** se utilizan como contadores para realizar el movimiento del huevo; **re.-** número de recogidas o lo que es lo mismo las veces que se ha pasado de una puerta a otra; **co.-** contador para el movimiento de cintas transportadoras; **t.-** variable del tiempo; **p.-** variable de los puntos; **pa,pae.-** contadores generales para la caida de los martillos; **m,n.-** coordenadas x e y de un martillo para su movimiento; **mb,nb.-** coordenadas de borrado del primer martillo; **g,h.-** coordenadas de borrado del segundo martillo; **gh,hn.-** coordenadas de borrado del segundo movimiento; **e,f.-** coordenadas de borrado del tercer martillo.

Desde la línea 97 a la línea 160 se realiza la impresión de toda la pantalla incluyendo Puntos y Record, sobre la que se desarrollará el juego.

Desde la línea 1000 a la línea 2000 aparece el bucle central del programa, en el cual se realiza el movimiento de todos los

```
3500 FOR a=1 TO 40: PRINT INK 7*
INT (RND*2);AT 17,y;"II";AT 18,y
;"II": OUT 254,a: BEEP .004,INT
(RND*30): NEXT a: BORDER 0: PRIN
T AT 17,2;"
          ";AT 18,2;"
                    "
3505 IF p>r THEN LET r=p
3510 PRINT INK 7; PAPER 2;AT 10,
11;"GAME OVER"
3520 PRINT INK 6; PAPER 1;AT 13,
8;"OTRA VEZ (S/N)"
3530 IF INKEY$="s" THEN CLS : GO
 TO 20
3540 IF INKEY$="n" THEN STOP
3550 GO TO 3530
8997 REM ***********************
8998 REM ** DATOS DE LOS UDGS **
8999 REM ***********************
9000 DATA 55,63,248,108,235,243,
63,46,119,255,136,85,85,34,255,2
38,116,252,159,55,54,31,252,236
9010 DATA 46,63,248,236,108,248,
63,55,238,255,136,85,85,34,255,1
19,236,252,159,54,55,31,252,116
9020 DATA 1,3,3,7,7,7,15,15,128,
192,192,224,160,224,240,240,15,1
5,7,3,15,30,24,12
9030 DATA 240,240,224,224,112,48
,56,60,15,15,7,1,1,1,1,1,240,240
,224,128,128,128,192,224
9040 DATA 1,3,3,7,5,7,15,15,128,
192,192,224,224,224,240,240,15,1
5,7,7,14,12,28,60
9050 DATA 240,240,224,192,240,12
0,24,48,15,15,7,1,1,1,3,7,240,24
0,224,128,128,128,128,128
9060 DATA 0,127,127,127,0,247,24
7,247,0,2,16,68,17,4,64,8,48,48,
48,48,48,48,254,248
```

A B C D E F G H I J K

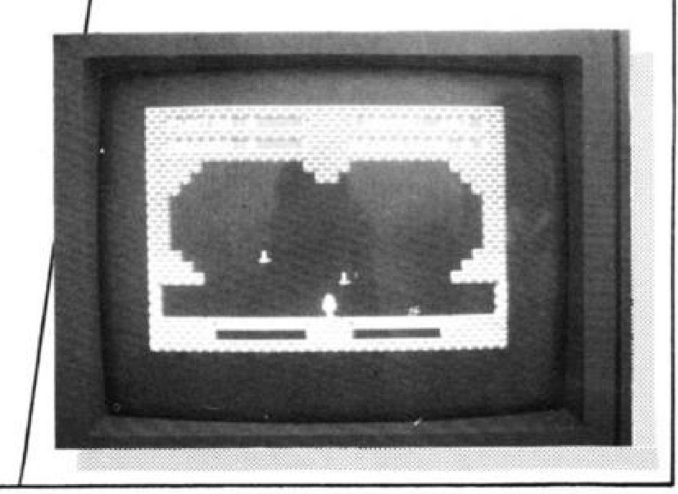
L M N O P Q R S T U

objetos, y viene dividido en secciones de la siguiente manera: desde la línea 1000 a la línea 1110 se efectúa la impresión y el movimiento de nuestro muñeco. Desde la línea 1200 a la línea 1245 se realiza el movimiento de las cintas transportadoras. Desde la línea 1300 a la línea 2000 o final del bucle se termina de realizar el movimiento de los tres martillos que caerán desde lo alto hasta el suelo.

Desde la línea 2097 a la línea 2150 se realiza la impresión y el cambio de puertas según las toque el muñeco.

Desde la línea 3000 a la línea 3040 se efectúa la comprobación de choques entre los martillos y nuestro huevo.

Desde la línea 3497 a la línea 3550 aparece el final del programa, donde se imprimirá en pantalla el clásico GAME OVER, y se dará la opción de volver a jugar otra vez o de dejar el juego.



BLANCO Y NEGRO UTILIDADES EN C/M



Original



Normal

LISTADO 1

```
        *D+
        *C-
        ORG     40000
        JP      NORMAL
        JP      INVER
PT1     DEFB    0,0
PT2     DEFB    0,0
TBL     DEFB    255,255,255
        DEFB    255,255,255
        DEFB    255,255,255
        DEFB    119,255,221
        DEFB    221,119,255
        DEFB    238,187,238
        DEFB    187,238,187
        DEFB    170,85,170
        DEFB    85,170,85
        DEFB    17,68,17,68
        DEFB    17,68,17,68
        DEFB    17,0,17,0
        DEFB    0,68,0,17
        DEFB    0,0,0,0
        DEFB    0,0,0,0
NORMAL  LD      DE,22528
        LD      HL,16384
FRAG    LD      B,3
        PUSH    BC
        PUSH    HL
BU1     LD      B,0
        PUSH    BC
        PUSH    HL
        LD      A,(DE)
```

Blanco y negro es una corta rutina en C/M que ocupa solamente 212 bytes, que transforma una pantalla en color en la equivalente en blanco y negro.

Esta rutina actua sobre una pantalla normal en color, transformando estos colores a tonalidades de grises. La ventaja que nos proporciona, es que podemos hacer un COPY en impresora, de pantallas en color, sin obtener los "fallos" que se ocultan tras los atributos.

La rutina tiene dos entradas



Invertida

BLANCO Y NEGRO

```
        PUSH  DE
        AND   7
        SLA   A
        SLA   A
        SLA   A
        LD    L,A
        LD    H,0
        LD    DE,TBL
        ADD   HL,DE
        LD    (PT1),HL
        POP   DE
        LD    A,(DE)
        PUSH  DE
        AND   56
        LD    L,A
        LD    H,0
        LD    DE,TBL
        ADD   HL,DE
        LD    (PT2),HL
        POP   DE
        POP   HL
        LD    B,8
        EX    DE,HL
        PUSH  DE
        PUSH  HL
BU2     PUSH  BC
        LD    A,(DE)
        XOR   255
        LD    HL,(PT2)
        LD    B,(HL)
        AND   B
        INC   HL
        LD    (PT2),HL
        PUSH  AF
        LD    A,(DE)
        LD    HL,(PT1)
        LD    B,(HL)
        INC   HL
        LD    (PT1),HL
        AND   B
        LD    B,A
        POP   AF
        OR    B
        LD    (DE),A
        INC   D
        POP   BC
        DJNZ  BU2
        POP   DE
        INC   DE
        POP   HL
        INC   HL
        POP   BC
        DJNZ  BU1
        POP   HL
        LD    BC,2048
        ADD   HL,BC
        POP   BC
        DJNZ  FRAG
        LD    HL,22528
        LD    BC,768
BU3     LD    (HL),56
        INC   HL
        DEC   BC
        LD    A,C
        OR    B
        JR    NZ,BU3
        RET
INVER   CALL  NORMAL
        LD    HL,15384
```

posibles, una en la dirección a partir de la cual se ensambla, en nuestro caso 40.000, y que produce el efecto descrito, y la otra, tres posiciones de memoria más adelante que hace lo mismo, ofreciendo el resultado invertido, es decir los puntos que estaban de tinta quedan como puntos de papel y los de papel como puntos de tinta.

Si deseáis ver la rutina en acción, tendréis que introducir el listado 1 con un ensamblador, o si no disponéis de esta herramienta, os proporcionamos un cargador BASIC que realiza la misma operación en el listado 2.

—Pero... ¿Cómo funciona?

—La transformación se realiza carácter a carácter. Para ello se emplean tres bucles, el primero, es el que indica el tercio de la pantalla. El segundo, recorre los 256 caracteres de cada bloque (8 filas × 32 columnas). Es aquí donde se toma el atributo correspondiente de la pantalla, se separa el color del papel y el de la tinta, y se almacenan en PT1 y PT2 los punteros que se calculen de las mascaras de color. El tercer bucle es el que transforma cada fila de carácter (1 byte) empleando las más-



```
1060
1070 BU4     LD    BC,6144
1080         LD    A,(HL)
1090         XOR   255
1100         LD    (HL),A
1110         INC   HL
1120         DEC   BC
1130         LD    A,C
1140         OR    B
1150         JR    NZ,BU4
             RET
```

LISTADO 2 -Cargador-

```
10 REM * CARGADOR C/M * COPY *
20 REM
30 CLEAR 39999: RESTORE 9000
40 LET S=0: FOR A=4E4 TO 40211
50 READ B: LET S=S+B: POKE A,B
60 NEXT A
70 IF S<>21819 THEN PRINT "ERR
OR": STOP
80 REM
90 REM * GRABACION *
100 REM
110 SAVE "COPY"CODE 40000,212
120 REM
130 REM * DEMOSTRACION *
140 REM
150 FOR A=0 TO 7: PRINT PAPER A
;: NEXT A
160 RANDOMIZE USR 40000
170 PAUSE 0
180 RANDOMIZE USR 40003
190 STOP

9000 DATA 195,138,156,195,0,157,
0,0,255,255,255,255,255,255,
255,119,255,221,255,119,255,
9010 DATA 255,238,[illegible],187,23
8,187,238,187,170,85,170,85,170,
85,170,85,17,68,17,68,17,68,17,6
9020 DATA 17,68,0,68,17,0,17,68,
0,68,0,17,0,68,0,17,0,0,0,0,0,0,
0,0,17
9030 DATA 0,88,33,0,64,6,3,197,2
29,6,0,197,229,26,213,230,7,203,
39,203,39,203,39,111,38
9040 DATA 0,17,74,156,25,34,70,1
55,209,26,213,230,56,111,38,0,17
,74,156,25,34,72,156,209,225
9050 DATA 6,6,235,213,229,197,26
,238,255,42,72,156,70,160,35,34,
72,156,245,26,42,70,156,70,35
9060 DATA 34,70,156,160,71,241,1
76,16,20,193,16,224,209,19,225,3
5,193,16,173,225,1,0,8,9,193
9070 DATA 16,161,33,0,88,1,0,3,5
8,58,35,11,121,176,32,246,201,20
5,138,156,33,0,64,1,0
9080 DATA 24,126,238,255,119,35,
11,121,176,32,246,201,0
```

caras de color, en el byte resultado.

La operación realizada es la siguiente; del puntero de tinta (PT1) se extrae la máscara de tinta y se realiza un AND con el byte de pantalla. Luego hacemos lo mismo con el papel (para saber que puntos de papel son los coloreados, invertimos el byte). Una vez tenemos el nuevo aspecto de la tinta y el de papel, los unimos con OR para obtener el byte que sustituirá al original en la pantalla.

Tras completarse los tres bucles, se cambian todos los atributos de la pantalla al color: papel 7, tinta 0, brillo 0, flash 0.

Si se desean cambiar las tramas que definen cada color, habrá que modificar los datos contenidos en las líneas 80 a 270 del listado ensamblador. Cada 8 datos conforman la trama de cada color, como si de un U.D.G. se tratase. El orden en que se almacenan es del más oscuro al más claro, del Negro (0) al Blanco (7).

La única desventaja, es que al simular los colores mediante tramas de grises, pequeños detalles con poco contraste de color se pierden. Aún así esperamos que os sea de utilidad.



LIBRERIA

FICHEROS EN BASIC

Autor: C. Delanoy. Editorial: Paraninfo. Páginas: 173



Esta obra les resultará muy útil a todos aquellos usuarios que quieran aprender a crear sus propios ficheros, partiendo de unos conocimientos mínimos o que quieran profundizar en este interesante tema de la gestión.

En el libro se ofrecen una serie de explicaciones sobre fichero manual e informático, y dentro de éste el acceso secuencial, directo e indexado.

En los Ficheros Secuenciales, se da una amplia información sobre cómo se crea un fichero directorio, profundizando en conceptos tales como grabación de información, la lectura, los modos separadores, etc... Todo ello acompañado de una gran variedad de ejemplos de diversa complejidad.

En sucesivos capítulos se explican los ficheros de acceso directo, los indexados, así como el modo de crearlos y ventajas y desventajas con respecto al secuencial.

En todo caso, habrá que tener en cuenta que los programas demostrativos han sido escritos en un Basic muy general y que algunos por tanto habrán de ser adecuados al Basic del Spectrum.

EL ORDENADOR EN LA EDUCACION BASICA

Autor: Tony Mullan. Editorial: Gustavo Gili. Páginas: 165





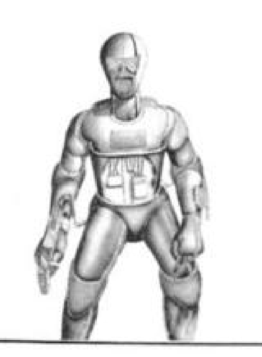
El libro está dirigido a padres y profesores que se encuentran en la necesidad de dar respuestas a problemas relacionados con los microordenadores en el ámbito de la Educación General Básica.

Así pues, se pretende dotar a los educadores de unos conocimientos mínimos con fines educacionales, y sobre todo, mentalizarles en la necesidad de que sus hijos y/o alumnos aprendan a programar.

La obra comienza con una introducción al ordenador, explicando someramente su historia, su rápido desarrollo y algunas de sus características más significativas, y continúa con unas explicaciones acerca de la importancia del ordenador en la actualidad, como herramienta indispensable de trabajo y estudio en las clases.

En otros capítulos trata en profundidad sobre el software, sobre su importancia a nivel educacional, sobre los modos de utilización de los programas, afin de sacarle el mayor rendimiento posible, siempre desde un punto de vista educacional, y sobre cómo han de ser planteados los ejercicios a los alumnos para ayudarles a introducirse de forma coherente y grata en el mundo del ordenador.





BOLETIN DE SUSCRIPCION

TODO UN AÑO DE PROGRAMAS E INFORMACION

POR SOLO **2000** Pts.

SUSCRIBETE AHORA Y OBTENDRAS

Ahorro de 400 Pts. SOBRE EL PRECIO DE VENTA EN TU KIOSCO un equivalente a 12 revistas por el precio de 10. Además tienes la seguridad de no perderte ningún número, aunque se agote.

Deseo que me envíen los doce próximos números del Mundo Spectrum por sólo 2.000 Pts. ahorrándome 400 Pts. sobre el precio de portada.

El primer número que deseo recibir es el ....

NOMBRE .... APELLIDOS ....

DOMICILIO .... CODIGO POSTAL ....

POBLACION .... PROVINCIA .... TELEFONO ....

FORMA DE PAGO:

- ☐ Contra reembolso al recibir el primer número.
- ☐ Giro postal nº.......
- ☐ Cheque adjunto nº.......

FOTOCOPIA, COPIA O RECORTA ESTE CUPON Y ENVIALO A GENESIS PUBLICACIONES, Tomás López, 3 - 6° - 28009 MADRID



¡Tu cinta para grabar guay!
HOLIDAY
HOLIDAY C-40
HOLIDAY C-60
HOLIDAY
Tu cinta virgen de 40, 60 y 90
HOLIDAY C-90
HOLIDAY
V. CARRERA
Fabricada por IBEROFON, S.A.
Avenida de Fuentemar, 35 - Polígono Industrial de Coslada - MADRID
Teléfonos 671 22 00-04-08-12 - Télex 42797 FONO E - Telefax (91) 671 39 09
Direct Metal DMM Mastering
COMPACT DISC DIGITAL AUDIO
discoflex


SPECTRUM • AMSTRAD CPC • COMMODORE • MSX
FREDDY HARDEST
FICHA TECNICA
• Dominio de las artes marciales: Patada - Puñetazo - Protección.
• Turbo-Laser de carga atmosférica.
• Salto controlable en altura y dirección.
• 30 Pantallas de Multiscroll.
• 64 Pantallas conectadas por ascensores y túneles.
• 16 Terminales de ordenador para hacer de Hacker.
• Enemigos inteligentes que se protegen de tu Laser.
• 5 Espécies de animales hostiles.
• 3 Tipos de Androides asesinos.
FX DOBLE CARGA
Signfica dos programas en uno, el doble de acción, el doble de aventura, el doble de gráficos y máxima adictividad.Prueba un FX DOBLE CARGA de DINAMIC, te sorprenderá.
SPECTRUM
COMMODORE
AMSTRAD CPC
MSX
FREDDY HARDEST, agente secreto y refutado playboy, se encuentra en un satélite enemigo con su nave averiada. Para escapar, deberá llegar hasta la base enemiga y robar un caza.
FREDDY, demostrará su destreza saltando, agarrándose a las argollas que encuentre, trepando por cuerdas, disparando su Laser,...
FREDDY, es todo un número uno, pero sobre todo, quiere escapar vivo.
875
DINAMIC
DINAMIC SOFTWARE. PZA. DE ESPAÑA, 18. TORRE DE MADRID, 29-1. 28008 MADRID. TELEX: 44124 DSOFT-E
TIENDAS Y DISTRIBUIDORES: (91) 314 • 18 • 04. PEDIDOS CONTRA REEMBOLSO: (91) 248 • 78 • 87.